



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

VÝUKOVÝ MODEL BALANCUJÍCÍ PLOŠINY

EDUCATIONAL MODEL OF A BALANCING PLATFORM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Tomáš Lička

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Martin Appel

BRNO 2021

Zadání bakalářské práce

Ústav: Ústav mechaniky těles, mechatroniky a biomechaniky
Student: **Tomáš Lička**
Studijní program: Aplikované vědy v inženýrství
Studijní obor: Mechatronika
Vedoucí práce: **Ing. Martin Appel**
Akademický rok: 2020/21

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Výukový model balancující plošiny

Stručná charakteristika problematiky úkolu:

Tato práce se bude zabývat vývojem výukového modelu balancující roviny. Zařízení bude pomocí motorů naklánět průhlednou desku ve dvou osách. Pohybem v těchto osách bude zařízení ovládat kuličku ležící na této plošině. Poloha kuličky bude snímána pomocí kamery, umístěné pod deskou. Kamera pomocí segmentace obrazu určí polohu. Výsledný model bude sloužit pro výuku regulace a programování v oboru Mechatronika. Vytvořený model musí být robustní, levný a jednoduchý na výrobu tak, aby se zařízení dalo případně vyrobit v takovém počtu, aby mohlo sloužit při výuce.

Cíle bakalářské práce:

- 1) Vytvořit simulační model balancující plošiny v nástroji Simulink.
- 2) Navrhnout více variant konstrukcí a po konzultaci s vedoucím práce vyrobit funkční prototyp výukového modelu.
- 3) Navrhnout řídicí jednotku pro řízení motorů a komunikaci s PC pomocí USB.
- 4) Vytvořit podklady pro výukovou úlohu s použitím vytvořeného zařízení.

Seznam doporučené literatury:

CORKE, Peter I. Robotics, vision and control: fundamental algorithms in MATLAB. Berlin: Springer, 2011. Springer tracts in advanced robotics, v. 73. ISBN 9783642201448.

GREPL, Robert. Kinematika a dynamika mechatronických systémů. Brno: Akademické nakladatelství CERM, 2007. ISBN 978-80-214-3530-8.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2020/21

V Brně, dne

L. S.

prof. Ing. Jindřich Petruška, CSc.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Abstrakt

Tato bakalářská práce se zabývá návrhem výukového modelu balancující plošiny se 3 stupni volnosti. Pomocí kamery ve spodní části se získává pozice míčku, ze které jsou počítány kroky pro krokové motory. V řešeršní části jsou popsány různé varianty řešení, které již byly vyrobeny. Zároveň jsou uvedeny výhody a nevýhody daných řešení. V následující části je popisováno zařízení, které je výstupem této bakalářské práce. Je popsána mechanická konstrukce, elektrická část a také hlavní myšlenky programu řídicího toto zařízení.

Klíčová slova

Balancující plošina, 3 stupně volnosti, regulace, zpracování obrazu, Matlab

Abstract

This bachelor thesis deals with the proposal of educational model of a balancing platform with 3 DOF. Camera placed in lower part of construction scan position of the ball which is used for calculation of steps of stepper motors. Various methods of solution with its advantages and disadvantages have been described in research part of this bachelor thesis. The solution that came up from this thesis is described in the next part. Mechanical construction, electric part and main parts of program have been shown.

Key words

Balancing platform, 3DOF, regulation, image processing, Matlab



Bibliografická citace

LIČKA, Tomáš. *Výukový model balancující plošiny* [online]. Brno, 2021 [cit. 2021-05-04]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/132637>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav mechaniky těles, mechatroniky a biomechaniky. Vedoucí práce Ing. Martin Appel.



Čestné prohlášení

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením Ing. Martina Appela s použitím informačních zdrojů uvedených v seznamu.

V Brně 20. května 2021

.....

Podpis



Poděkování

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Martinu Appelovi za ochotu a cenné rady při zpracovávání této závěrečné práce. Zároveň bych rád poděkoval rodině a přítelkyni za motivaci a podporu při tvorbě práce.

Obsah

Úvod.....	11
1 Rešerše.....	12
1.1 Pohyb konstrukce.....	12
1.2 Snímání pozice míčku.....	13
1.3 Pohon	14
2 Realizace.....	15
2.1 Mechanická konstrukce	15
2.1.1 Požadavky na konstrukci	15
2.1.2 Kinematický rozbor	16
2.1.3 Řešení ramen	17
2.2 Elektronika.....	18
2.2.1 Arduino	18
2.2.2 Krokové motory.....	20
2.3 Program.....	22
2.3.1 Schéma programu	22
2.3.2 Serial port.....	23
2.3.3 Zpracování obrazu	24
2.3.4 Naklánění roviny	26
2.3.5 Matematický model	27
2.4 Grafické prostředí	28
3 Úloha pro studenty	30
3.1 Zadání	30
3.2 Zpracování obrazu	30
3.3 Transformační matice	31
3.4 Řízení	32
4 Výsledky.....	35
4.1 Přesnost náklonu	35
4.2 Ukázky regulace.....	36
4.3 Výsledný vzhled.....	38
5 Přehled nákladů	40
Závěr.....	41
Seznam použité literatury.....	42

Seznam použitých symbolů	44
Seznam obrázků	46
Seznam tabulek.....	47
Seznam příloh	48

Úvod

Balancování míčku na plošině je častá úloha při řízení. Při pokusech o stabilizování míčku je třeba uvažovat pohybové rovnice (diferenciální rovnice), které popisují daný model a zároveň uvažovat nedokonalosti způsobené prací na reálném zařízení. Úloha řízení pozice míčku nemusí být vztažena pouze k jednomu bodu, ale může být časově proměnná tak, aby míček tvořil jistou trajektorii. Příkladem může být kružnice.

Analogií mechanismu vyvíjeného v rámci této bakalářské práce jsou drony. Výrobky v posledních pár letech našly uplatnění jak mezi amatéry, tak mezi hasiči, policií a armádou. Ti je využívají pro záchranné a monitorovací účely. Nesmírnou výhodou používání dronů je větší rozhled než při pěším průzkumu a je možné na ně připevnit například termovizi pro snadnější hledání osob. Jedná se zároveň o mnohem ekonomičtější variantu než používání helikoptér [1].

Vítr, který fouká do dronu ze všech stran, se dá chápat jako pohyb míčku na plošině. Aby dron zůstal na stejné pozici, musí na vítr reagovat a posílat jiný výkon do každého motoru, což se dá představit jako naklánění plošiny.

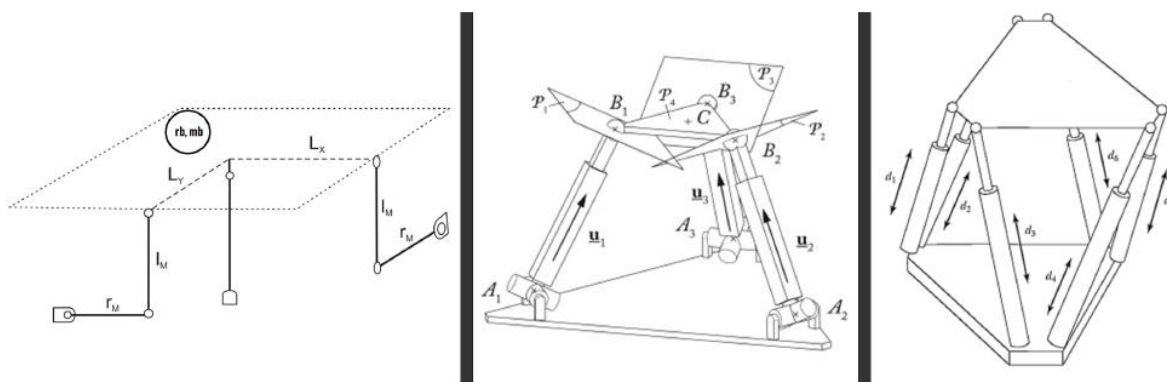
Cílem této bakalářské práce je tedy navrhnout konstrukci a naprogramovat reálný mechanismus, který bude balancovat s míčkem, jehož poloha bude snímána kamerou umístěnou ve spodní části konstrukce. Dalším cílem je vytvořit matematický model v nástroji Simulink pro porovnání s reálným mechanismem. Posledním cílem je vytvoření úlohy pro studenty nižších ročníků. V programu se nachází několik částí, které byly s úmyslem vytvořeny jinak a složitěji než v popisované úloze, aby studenti museli pochopit kód a byli schopni ho upravit.

1 Rešerše

V této části budou popsány různé varianty řešení vývoje mechanismu. Nejdříve budou popsány možnosti pohybu konstrukce. Rozebrány budou 3 základní možnosti. V další části budou rozepsány možnosti snímání míčku. Možností je opět více a je důležité zvolit vhodnou metodu vzhledem k celkové konstrukci. Poslední částí této kapitoly jsou varianty pohonu plexiskla. Při vývoji mechanismu byla vybrána jedna z kombinací popisována v této kapitole.

1.1 Pohyb konstrukce

Rozbor pohybu konstrukce je možné provést z pohledu počtu stupňů volnosti. Na obrázku 1 jsou zobrazeny tři základní typy konstrukcí, každý z nich má více variant provedení. V následujících odstavcích budou jednotlivé varianty blíže popsány i co se týče jejich výhod a nevýhod.



Obrázek 1 Více variant mechanismu [2]

- **2 stupně volnosti:** Mechanismus s tímto řešením je vidět na levém schématu. Aby byla konstrukce stabilní a nepadala vlivem své vlastní tíhy, tak by musela být v jednom místě trvale podepřena, jak je naznačeno na schématu. V tomto místě by poté byla sférická vazba a plošina by se nakláněla pomocí dvou motorů.

Výhodou tohoto řešení je zajištění translační nepohyblivosti plošiny ve vodorovných osách. Zároveň je i úspornější co se týče počtu motorů. Nevýhodou při využití kamery by však mohlo být, že by trvalá podpora mohla zabraňovat v jejím výhledu.

- **3 stupně volnosti:** Volba, která byla zvolena pro mechanismus popisovaný v této práci. Na obrázku 1 uprostřed je schéma, které má 3 stupně volnosti, s konstrukcí založenou na lineárních motorech. Tato volba umožňuje umístění kamery pod plexisklo, protože jí nic nebrání ve výhledu. Je zde ovšem potřeba o jeden motor víc. Základny motorů by musely být uchyceny pomocí sférických vazeb.

Konkrétní firmou, která se zabývá výrobou podobných mechanismů, je například polská firma Motion System [3] fungující od roku 2009. Konstrukce mechanismu popisovaného v této bakalářské práci je řešená stejným způsobem jako jejich výrobek s označením PS-3TM-200, který je vidět na obrázku 2. Zobrazený výrobek je robustnější, protože má unést až dvousetkilogramové závaží, přičemž sám váží pouze 100 kg.

Pro provedení se 3 stupni volnosti je možné také použít 4 motory. Uchycení na plexiskle by bylo realizováno opět pomocí sférických vazeb.

- **6 stupňů volnosti:** Zařízení se 6 stupni volnosti se nazývá Stewartova plošina. V praxi se tato plošina používá ke konstrukci leteckých a automobilních simulátorů. S použitím 6 stupňů volnosti souvisí i možný pohyb plexiskla ve vodorovné rovině, což není požadovanou vlastostí u mechanismu vyvíjeného v této práci. Aby se plexisklo traslačně nepohybovalo, je potřeba dopočítávat u všech šesti lineárních motorů zobrazených na schématu jejich vysunutí. Jedná se o technicky pokročilé a elegantní řešení, ale nevýhodou je finanční a výpočetní náročnost.



Obrázek 2 Výrobek PS-3TM3-200 [3]

1.2 Snímání pozice míčku

Velmi podstatnou částí tohoto projektu je určení způsobu snímání pozice míčku. V úvahu připadají dvě řešení:

- **Dotykový displej:** V případě použití dotykového displeje, jako snímače polohy míčku, by mohl být celý mechanismus naprogramovaný pouze v Arduino. Signál z dotykového displeje se dá zpracovávat přímo v něm a jeho napájení by se řešilo bateriemi. Krokové motory by stále musely být napájeny externím zdrojem. Tento způsob řešení má ale i nevýhody.

Dotykové displeje se rozdělují na kapacitní a odporové. Pro určování polohy míčku je vhodnější odporový displej, neboť kapacitní by mohl způsobovat šum. Míček by musel mít větší hmotnost, aby ho odporový displej zaznamenával. Nevýhodou tohoto způsobu snímání je nutnost neustálého kontaktu míčku s displejem, jinak by byla ztracena informace o jeho pozici. Jestliže motory nakloní plexisklo příliš rychle, je možné, že míček poskočí. Při použití druhého způsobu řešení, kamery, tento problém nenastává.

- **Kamera:** Při použití kamery je třeba zajistit, aby viděla celou plochu plexiskla. Jelikož je konstrukce relativně nízká, tak je vyžadován velký zorný úhel kamery jak v horizontálním směru, tak vertikálním.

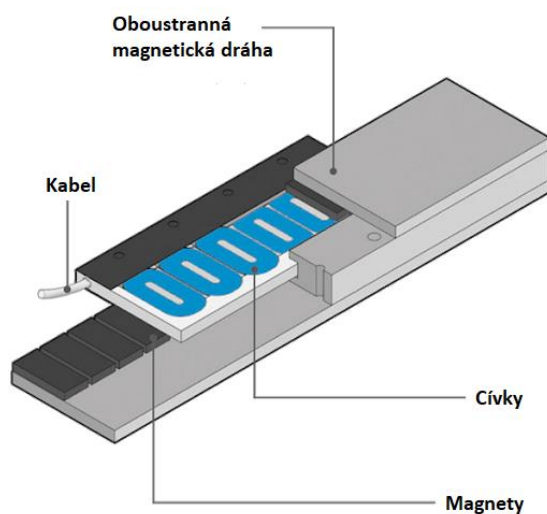
Umístění kamery je možné i nad deskou, která stabilizuje míček. V takovém případě by celá konstrukce byla nižší. Nad konstrukcí by však muselo vyčnívat rameno s upevněnou kamerou. Toto řešení je jednodušší variantou, co se týče zpracování obrazu, neboť deska nemusí být průhledná a může zajišťovat dostatečný kontrast oproti míčku.

1.3 Pohon

Důležitou částí je výběr pohonu plošiny kvůli jeho vysokým nárokům na přesnost a rychlost, přičemž se jedná o nejnákladnější položku konstruovaného modelu.

- **Lineární motory:** V praxi se hojně využívají lineární pohony, které jsou velice přesné. Jediným rozdílem mezi motory konajícím rotační pohyb a lineárními je ten, že stator netvoří kruh, ale je rozvinutý do roviny. To umožňuje pohyb posuvné části jako translaci, nikoli rotaci [4].

Tento způsob řešení má výhody v úspoře prostoru. Usnadňuje konstrukci strojů, u kterých je požadován translační pohyb a je schopný i při velmi malých posuvech tlačit velkou silou proti závaží. Své uplatnění našly lineární motory například u balících strojů, polohovacích systémů a výrobních linek [4].



Obrázek 3 Řez lineárním motorem [4] - upraveno

Řez lineárním motorem je popsán na obrázku 3. Cívky jsou namotané v rovině, a díky přívodu napětí na cívky se permanentní magnet pohybuje.

Při použití tohoto typu pohonu by bylo vhodné zpětnovazebně kontrolovat jeho pozici.

- **Krokové motory:** Jedná se o způsob pohonu, který byl použit v tomto projektu. Krokové motory jsou podrobněji popsány v kapitole 2.2.2.
- **Servo motory:** Důležitou vlastností tohoto typu pohonu je možnost přesné regulace polohy, rychlosti nebo momentu. V závislosti na aplikaci se může regulovat pouze jedna zmiňovaná veličina [5]. Jde o typ pohonu, který má velice plochou zatěžovací charakteristiku [6]. Výhodou servo motorů oproti krokovým motorům je i nižší napájecí napětí.

2 Realizace

V následující části bude popsán postup vývoje mechanismu. Bude rozebrána jeho mechanická konstrukce, která se skládá ze spodní části, ve které je uložena kabeláž, a ze svislých stojen, které podpírají prstenec. Na něm jsou přimontovány krokové motory. Dále bude ukázána použitá elektronika, jejíž základem je Arduino Mega 2560. Poslední probíranou částí bude hlavní myšlenka programu, který řídí celý mechanismus.

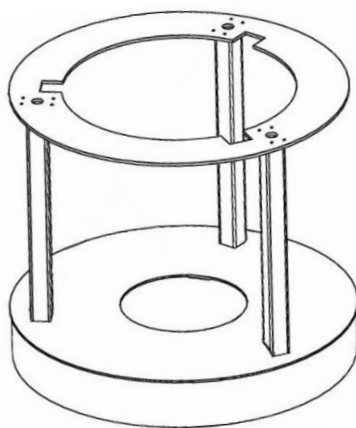
2.1 Mechanická konstrukce

Hlavním materiálem konstrukce je nerez z důvodu dlouhé životnosti a pevnosti. Pro ramena by byl nerez moc těžký, proto byla vyrobena z duralu. Díky tomu jsou srovnatelně těžká jako při vytisknutí 3D tiskárnou, ale zajišťují pevnost. Délka ramen byla zvolena kompromisem tak, aby poskytovala dostatečný rozsah pohybu a zároveň aby měly motory dostatečný moment pro manipulování nimi.

2.1.1 Požadavky na konstrukci

Na konstrukci, jakožto kostru celého mechanismu, byly kladeny následující nároky.

- **Robustnost:** V programu je pozice kamery a výchozí pozice všech motorů pevně dana. Vždy se musí startovat z té stejné pozice, jinak program nebude fungovat správně. Z toho plyne, že veškeré deformace musí být zanedbatelné, ideálně nulové. Je nutné také zajistit, aby se v důsledku pohybu motorů nepohybovaly stojny, což je nežádoucí stav.
- **Malá hmotnost:** Mechanismus je konstruovaný tak, aby sloužil pro účely studentů a mohli si ho odnést i z laboratoře. Proto je potřeba, aby nebyl zbytečně těžký, což by zhoršilo jeho mobilitu. Plexisklo se dá navíc odebrat díky tomu, že vazba s ním je rozebíratelná. Tím se rozměry na přenos ještě zmenší a horní rameno z obrázku 5 bude viset dolů. Zároveň se dá rozebrat i plech se stojnami od spodní části. Tyto dvě části jsou spojeny šrouby, aby byl v případě potřeby možný přístup ke kameře nebo Arduino.
- **Bezpečnost:** Dalším důležitým faktorem je bezpečnost. Na konstrukci není žádná ostrá hrana a ani nehrozí úraz elektrickým proudem způsobený zdrojem, který se nachází ve spodní části konstrukce. Byly instalovány i dva vypínače, které odpojují danou část obvodu od sítě.
- **Uspořádání kabelů:** Ve spodní části konstrukce se nachází veškerá kabeláž, která je potřeba pro fungování mechanismu. Kabely jsou vedeny po okraji spodní části a k motorům jsou vedeny vertikálními stojnami sloužícími i jako podpora pro motory.
- **Dostupnost:** Konstrukce je vyrobena z kovu pro větší tuhost. Méně nákladnou variantou by bylo vytisknutí na 3D tiskárně, ale tato možnost není vhodná pro ramena pohybující plexisklem. Některé komponenty spodní části konstrukce by ale bylo možno vyrobit tímto způsobem za účelem snížení nákladů. Celkové náklady za kompletní mechanismus jsou uvedeny v tabulce 4 v kapitole 5. Pro účely této práce byla konstrukce vyrobena zdarma, ale pro sériovou výrobu by stálo za zvážení právě využití 3D tiskáren.
- **Estetičnost:** Posledním požadavkem je vzhled celé konstrukce. Většina svárů je provedena tak, aby nebyla vidět a ostatní spoje nejsou viditelné, aby nekazily vzhled. Náčrt konstrukce je zobrazen na obrázku 4.

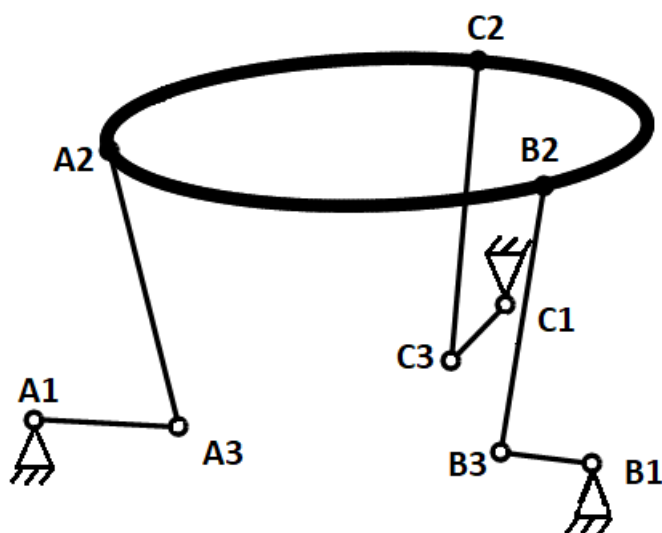


Obrázek 4 Nákres konstrukce

2.1.2 Kinematický rozbor

Schématický nákres mechanismu je popsán na obrázku 5. Všechna tělesa jsou uvažována jako tuhá a není předpokládána jejich deformace v jakémkoli směru. Zároveň je zanedbána hmotnost těles, jejich momenty setrvačnosti a odpory, které by mohly klást.

Na obrázku 5 je označeno 6 rotačních vazeb A1, B1, C1, A3, B3, C3. První tři rotační vazby jsou hřídele motorů, na které jsou nasunuty spodní ramena. Další tři rotační vazby jsou spojnice spodních a horních ramen.



Obrázek 5 Schéma mechanismu

Níže je proveden kinematický rozbor mechanismu, pro dokázání jeho stupňů volnosti. Pro rozbor bylo použito schéma na obrázku 5. Každé těleso v prostoru má 6 stupňů volnosti, posuvy ve všech třech osách a rotaci kolem každé z nich. Mechanismus obsahuje:

- 8 těles (n_t)- základna, 3x dolní rameno, 3x horní rameno a horní deska z plexiskla
- 9 kinematických dvojic (j)

Z těchto znalostí je možné dosadit do vztahu (1) pro výpočet stupňů volnosti i

$$i = 6(n_t - j - 1) + \sum_i f_i = 3. \quad (1)$$

Člen f_i vyjadřuje počet dovolených stupňů volnosti jednotlivými vazbami. Rotační vazby v prostoru dovolují rotaci kolem jedné osy, sférické kolem všech tří os.

Mechanismus má tedy 3 stupně volnosti, ve kterých může operovat. Prvním je translace podél svislé osy z, čehož se využívá při inicializaci programu, aby se plošina dostala do pracovní pozice. Posuv ve svislé ose může být ještě 5 cm výš než je výchozí pracovní pozice. Druhým a třetím stupněm volnosti jsou rotace kolem vodorovných os x a y. Tyto pohyby umožňují naklánění desky do požadovaného úhlu, který je omezen programem. Náklon plošiny může být realizován až do 30 °. Používá se však rozpětí pouze do 10 °, i to je dostačující pro rychlou kontrolu pozice míčku.

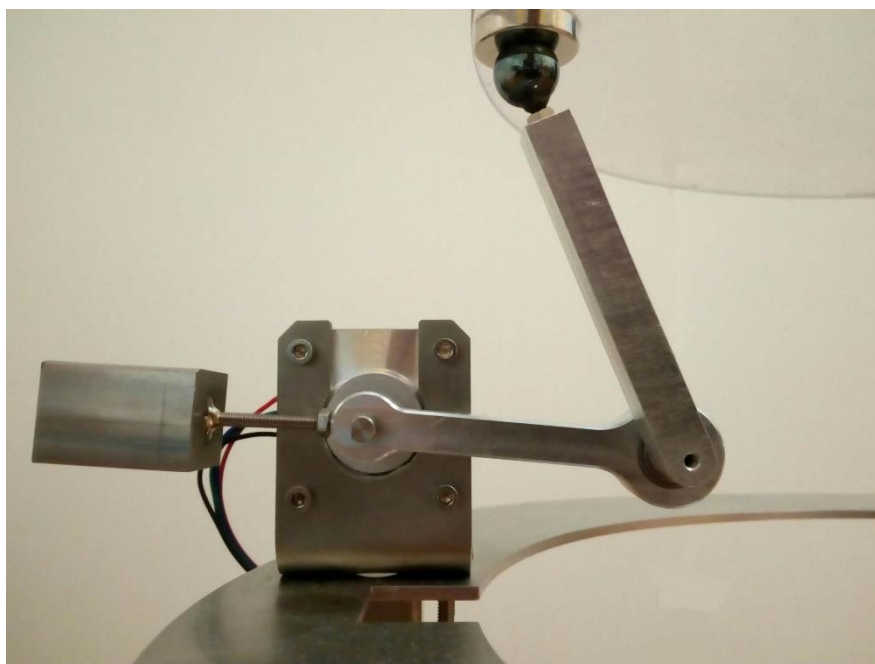
2.1.3 Řešení ramen

Sférické vazby mechanismu, které jsou na obrázku 5 označeny body A2, B2, C2, jsou na zařízení realizované kovovými kuličkami, které se přivařeným šroubkem připevní do horního ramene a zajistí dvojicí matek. Délka zašroubování je variabilní díky tomuto řešení. Protikusy k nim jsou magnetické a zajišťují tak soudržnost vazby. Protikusy pro kuličky jsou přilepeny k desce z plexiskla.

Realizace rotačních vazeb je vyřešena tak, aby měla dostatečnou tuhost a umožňovala rotaci pouze kolem jedné osy s minimální vůlí pohybu v jiných směrech. Na prototypu byly použity díly vytisknuté na 3D tiskárně, které tuto vlastnost nesplňovaly. Bylo možné s nimi otáčet i kolem jiných os, proto byly nahrazeny duralovými rameny.

Samotná vazba je poté tvořena ložiskem s minimálním odporem při rotaci v plném rozsahu. Ložisko je svým vnějším průměrem uchyceno ke spodnímu ramenu pomocí tří šroubků. Horní rameno je naopak spojeno s vnitřním kroužkem ložiska.

Horní rameno je ve své dolní části prodlouženo, aby prošlo i dutinou ložiska. Ve vnitřní straně tohoto prodloužení je závit, do kterého je zašroubován šroub držící celou vazbu semknutou k sobě. I toto řešení přináší nežádoucí vůli, s níž ideální rotační vazba nepočítá. Zde je způsobena tím, že k rozměrově malému ložisku jsou namontovány kusy dlouhé 6 a 8 cm. Na ložisko vzniká velká páka způsobující drobné posuvy, které jsou oproti předchozím vytištěným ramenům zanedbatelné. Jedná se o jednotky milimetrů, zatímco u vytištěných dílů se jednalo o centimetry.

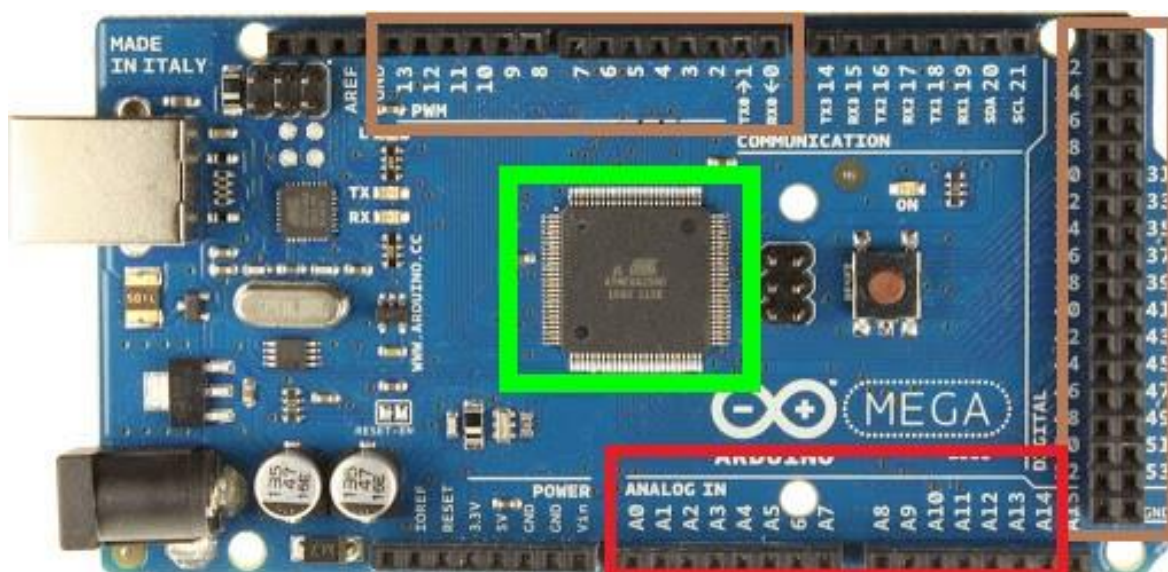


Obrázek 6 Soustava ramen

2.2 Elektronika

2.2.1 Arduino

Pro spojení mezi Matlabem a motory bylo zvoleno Arduino Mega 2560 [7], které komunikuje s počítačem pomocí serial portu. Úkolem Arduina je řídit motory elektrickými impulsy o přesně daný počet kroků správným směrem. Arduino je vybavené procesorem ATM Mega 2560, zároveň nabízí 54 digitálních pinů, 16 analogových a 4 kanály pro sériovou komunikaci.



Obrázek 7 Arduino Mega 2560 [7] - upraveno

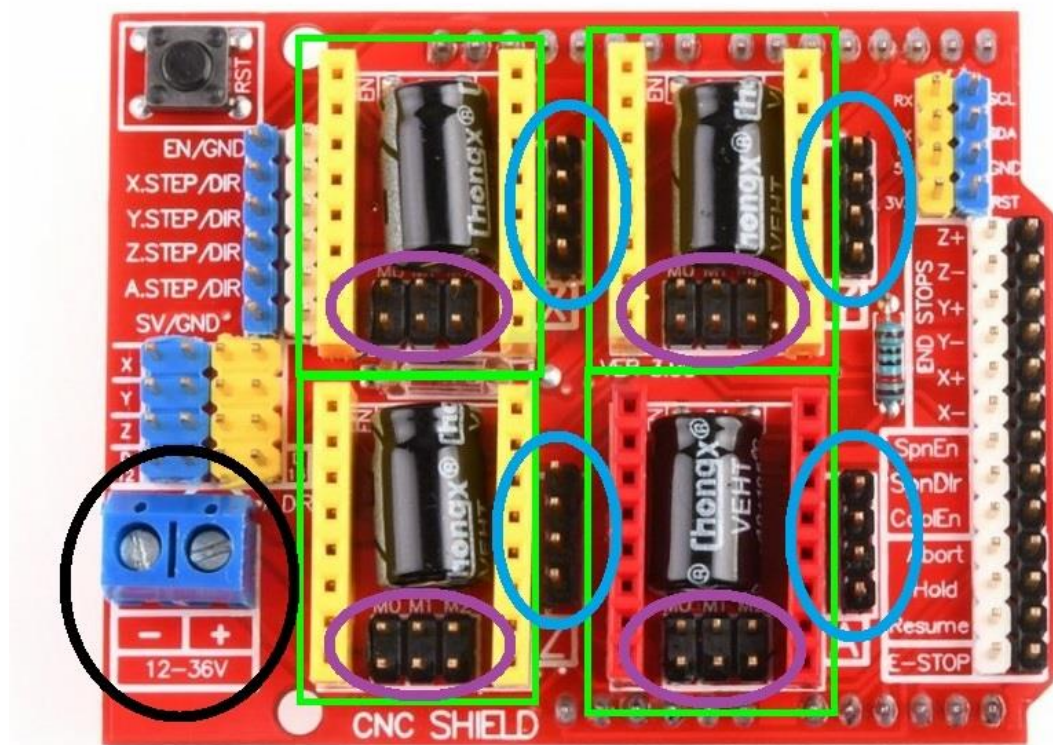
Na obr. 7 je zobrazeno použité Arduino. Hnědě jsou vyznačeny digitální porty, červeně analogové a v zeleném rámečku je samotný mikroprocesor.

K Arduinu byl dokoupen ještě CNC Shield, který je určen pro řízení krokových motorů. Je vybaven 4 sloty pro drivery a piny, na které se připojí krokové motory. Pomocí jumperů, které by spojily vhodné piny je možné docílit mikrokrokování s požadovanou hodnotou, a to až 1/64 kroku. CNC Shield na sobě má piny s názvy X.Stp a X.dir. Tyto dva piny poté i pro další dva motory, které jsou označeny písmeny Y a Z.

V programu Arduina poté stačí zajistit, aby byly správně nastaveny piny, které jsou připojeny k CNC Shieldu. V tomto případě pin X.stp odpovídá digitálnímu pinu 2 a X.dir digitálnímu pinu 5. V programu je už pouze nastavován na pin 2 logická 1 a logická 0 s malými pauzami, aby se cívky v motoru stihly nabít a posunuly tak hřídeli. Tato pauza byla pomocí experimentů stanovená na 3000 μ s. Na pinu 5 je po celou dobu pohybu pouze logická 0 nebo logická 1. Počet opakování tohoto děje je roven počtu kroků, o které se motor otočí.

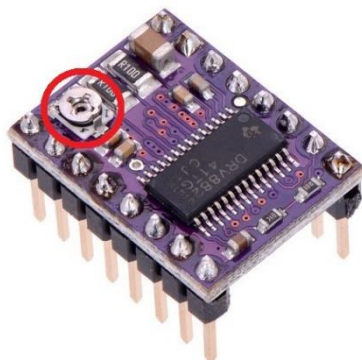
Aby se motory začaly pohybovat potřebují napětí 12-36 V. Takové napětí je dodáváno do CNC Shieldu pomocí svorek z externího zdroje, konkrétně 24 V, který je zapojen přímo do sítě.

Na obr. 8 je zobrazen CNC Shield [8]. Zeleně jsou vyznačeny 4 sloty pro drivery. Tyto sloty mají své odpovídající piny step a dir na spodní straně shieldu. Modře jsou poté označeny piny pro připojení motorů. Dva piny vždy odpovídají jedné cívce v krokovém motoru. V černém kolečku jsou svorky, na které se připojí externí zdroj napětí. Fialově jsou označeny piny, které umožňují mikrokrokování. Správným propojením těchto pinů je možné docílit až 64násobného zjemnění kroků.



Obrázek 8 CNC Shield [8] - upraveno

Jako driver byl použit DRV8825 [9], který je na obrázku 9. Pomocí otáčení potenciometru označeného červeně lze nastavovat proud protékající motory. Pro tento konkrétní typ driverů platí, že napětí na potenciometru se rovná polovině proudu, který projde až do motoru. Jelikož motory mají limit 1,3 A, je nutné mít toto napětí pod úrovní 0,65 V. Je však vhodné, aby motory protékal proud jen tak velký, aby měly dostatečný moment na manipulování s mechanismem. Příliš velký proud by způsoboval zahřívání pinu na driveru. Je nutné najít kompromis mezi těmito skutečnostmi.



Obrázek 9 Driver DRV8825 [10] - upraveno

V programu, který je nahrán do Arduina je funkce převzatá z diskuzního fora stackExchange [11]. Tato funkce zajišťuje přečtení textu, který je posílán Matlabem.

2.2.2 Krokové motory

Náklon plošiny je řízen pomocí tří krokových motorů Nema 17HS4401 [12]. Použité krokové motory patří do kategorie hybridních (kombinují vlastnosti aktivních a pasivních). Krokový motor je speciálním typem synchronního, který se řídí pomocí počtu kroků (200 na jednu celou otáčku) a směrem (1 nebo 0). Právě to, že se dá jedna jeho otáčka rozdělit na 200 kroků zaručuje jeho přesnost pohybu bez nutnosti regulace natočení. Dalšími výhodami podle [13] jsou:

- **Vyšší životnost:** Krokové motory ve své konstrukci neobsahují kartáče ani komutátor. Tyto dvě části se používáním opotřebovávají a vedou k menší účinnosti motoru.
- **Snadné ovládání rychlosti:** Rychlost tohoto typu motorů se řídí pomocí mezer mezi pulsy, které do něj přichází. Je však třeba vyhnout se pásmu rezonancí, v nichž by motor nepracoval správně.
- **Přetížení:** V některých aplikacích může cíleně docházet ke krátkodobému přetížení motorů. Pro krokové motory to stejně jako pro jakýkoli jiný typ motoru není ideální, ale není to pro ně fatální stav, dokážou se z něj vrátit bez poškození.

Kromě výhod pro použití tohoto typu motorů se vyskytují i nevýhody, které je potřeba brát na vědomí.

- **Točivý moment:** Točivý moment krokových motorů je výrazně menší než u stejnosměrných motorů podobných rozměrů. Malý točivý moment byl hlavním důvodem nainstalování protizávaží, které snižují moment na hřídeli motoru. Pracovní pozice spodního ramene je po většinu času vodorovná, takže působí na hřídel motoru maximálním momentem. Protizávaží byla napočítána tak, aby pokryla téměř čtvrtinu maximálního momentu motoru (přesněji 7 Ncm).

- **Napájení:** Použité motory musí být napájeny externím zdrojem s výstupním napětím v rozmezí 12 V až 36 V, zatímco ostatní typy motorů podobných výkonů jsou napájeny nižším napětím.
- **Rychlost:** Použití krokových motorů není vhodné pro aplikace, při nichž se vyžaduje vysoká rychlost otáčení. Při vyšších rychlostech ztrácejí svůj točivý moment.

Některé základní parametry použitých motorů převzaté z datasheetu [12] ukazuje tabulka 1.

Tabulka 1 Parametry motoru [12]

Úhel kroku [°]	Délka hřídele [mm]	Nominální proud [A]	Odpor rezistoru [Ω]	Induktance cívky [mH]
1,8	40	1,7	1,5	2,8
Statický moment [Ncm]	Zbytkový moment [Ncm]	Moment setrvačnosti [gcm]	Počet drátů	Hmotnost motoru [g]
40	2,2	54	4	280

Mikrokrokování

Při některých aplikacích je vhodné, aby se motor otočil o menší úhel, než je jeden plný krok. K těmto účelům slouží takzvané mikrokrokování. Je to proces, při kterém jsou použity všechny fáze najednou, avšak ne s plným výkonem. Tento proces nijak neovlivní přesnost chodu motoru, naopak ho ještě zpomalí. Rychlost impulsů vstupujících do motoru je konstantní, ale každý z nich způsobí otočení hřídele o menší úhel. Z toho plyne, že se celková rychlost otáčení zmenší.

Mikrokrokování tedy přináší výhody v přesnosti pohybu hřídele, na druhou stranu má i své nevýhody [14]. Čím větší zjemnění kroku, tím menší statický moment motor má. Tento moment klesá velkým tempem, jak ukazuje tabulka 2.

Při použití největšího možného mikrokrokování dovoleného CNC Shieldem (1/64) má motor méně než 2,5 % svého maximálního statického momentu, v tomto konkrétním případě méně než 1 Ncm. Byly použity 2 mikrokroky na jeden plný krok, jeden krok motoru nyní znamená otočení o 0,9 °. Tento postup byl nutný z důvodu velkého rozdílu náklonu plexiskla při použití plného kroku 1,8 °.

Mikrokrokování se realizuje propojením určitých pinů na CNC Shieldu (popsaném v kapitole 2.2.1) pomocí jumperů. Použití mikrokrokování je dalším důvodem přidání protizávaží.

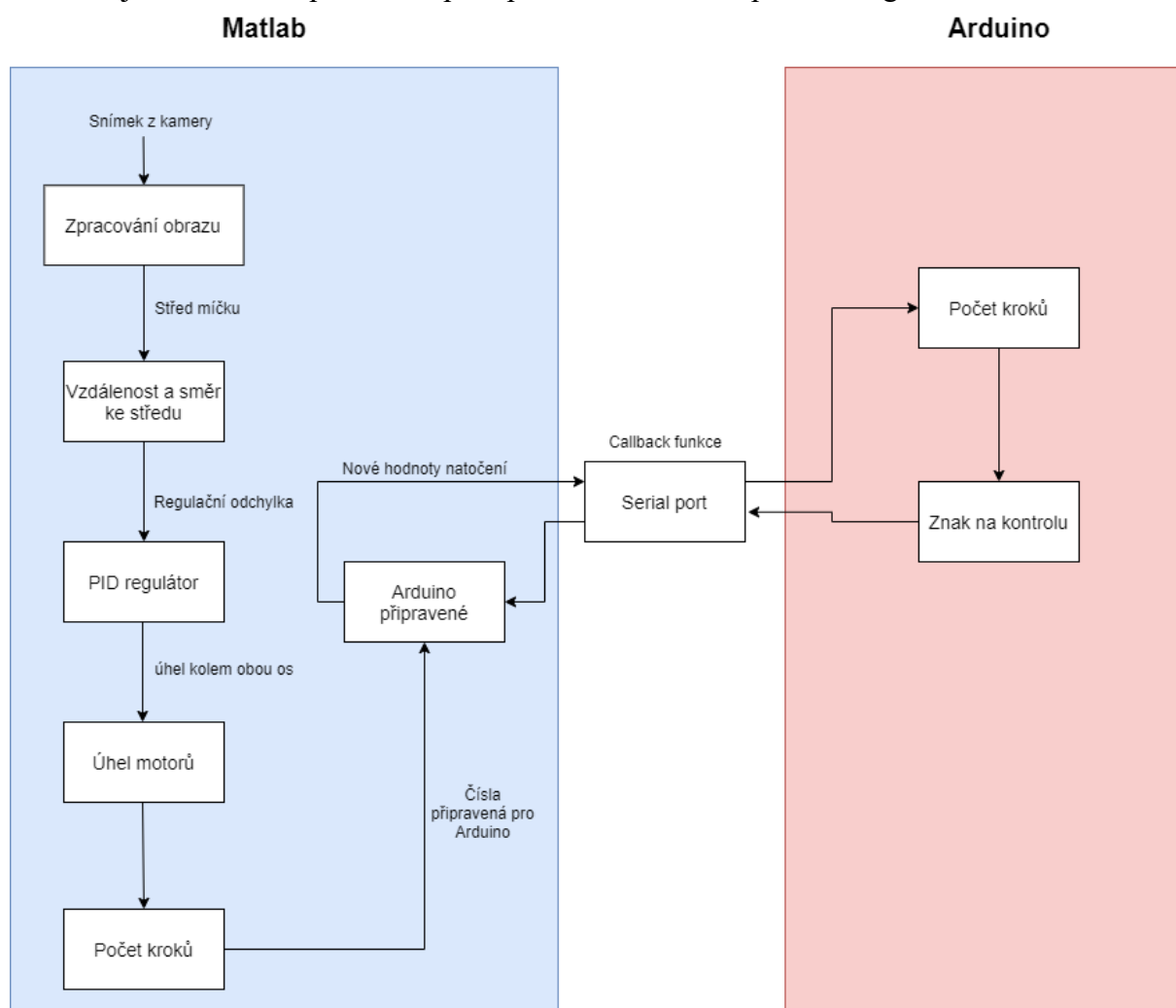
Tabulka 2 Ztráty výkonu mikrokrokováním [14] - upraveno

Počet mikrokroků na 1 plný krok	Statický moment [%]
1	100
2	70,71
4	38,27
8	19,51
16	9,8
32	4,91
64	2,45

2.3 Program

2.3.1 Schéma programu

Na obrázku 10 je zjednodušené schéma programu. Celý program začíná inicializací, zkontrolováním komunikace s Arduinem a připojením kamery. Další částí, která zde není zahrnuta je nastavování parametrů pro zpracování obrazu a pro PID regulátor.



Obrázek 10 Schéma programu

Vstupem do schématu je pořízení snímku kamerou. Tento snímek je dále zpracován pomocí *Image Processing Toolbox*. Proces zpracování je konkrétněji rozepsán v kapitole Zpracování obrazu (kapitola 2.3.3).

Po získání hodnoty v pixelech středu míčku v obou osách se zjistí vzdálenost od požadovaného bodu (v tomto případě středu plexiskla) pomocí vztahu (2).

$$x_p = \sqrt{S_b^2 - P^2}, \quad (2)$$

kde x_p je vzdálenost od požadovaného bodu v pixelech, S_b je střed míčku a P je hodnota požadovaného bodu v pixelech. Výsledkem vztahu 2 je dvousložkový vektor (provádí se v obou osách).

U míčku se zjišťuje nejen jeho poloha ale i jeho rychlost. K výpočtu se používá základní vztah (3).

$$v = \frac{dS_b}{dt}, \quad (3)$$

kde v je rychlost v jednotlivých osách, dS_b změna polohy v jednotlivých osách a dt je rozdíl časů. Vzorec (3) se sice používá pro průměrnou rychlost, ale jelikož čas mezi měřeními polohy je velmi malý, tak je možné ji prohlásit za okamžitou.

Aktuální poloha a rychlost jsou parametry, které vstupují do PID regulátoru. Výstupem z něj je dvojice úhlů, o které se má plexisklo natočit (jeden pro každou osu).

Princip výpočtu množství kroků jednotlivých motorů je popsán v kapitole 2.3.4. Po získání hodnot, které se už posílají do Arduina (počet kroků a směr pro každý motor), Matlab opakuje celý tento proces, bez ohledu na to, jestli byly aktuální hodnoty odeslány do Arduina. K posílání hodnot do Arduina slouží funkce *Callback function*, která pošle získané hodnoty do Arduina hned poté, co dostane „povolení“ od Arduina. Tento princip je blíže popsán v kapitole Serial port (2.3.2).

2.3.2 Serial port

Komunikace mezi Matlabem a Arduinem je jedním ze stěžejních pilířů, na kterém stojí správný chod mechanismu. Komunikace probíhá pomocí serial portu, což je dvoucestné komunikační rozhraní, které umožňuje posílat informace z Matlabu do Arduina a naopak, a to bit po bitu [15].

Každá informace, která má být přenášena serial portem, v tomto případě čísla, je zahájena start bitem. Je to bit, který má logickou 0 a uvádí celou zprávu. Následuje hlavní tělo zprávy. Po ní je možné zkontrolovat úplnost zprávy pomocí parity bitu. V tomto programu není kontrola parity použita.

Aby zařízení vědělo, kdy má zaznamenat, jestli přijímá logickou 1 nebo 0, je potřeba, aby modulační rychlost obou zařízení byla stejná. Modulační rychlost, často označovaná jako baud rate, je rychlost přenosu. Vyjařuje, kolik bitů se přeneso za jednu sekundu. V tomto případě je baud rate zvolena 2 000 000.

Pro větší jistotu zaznamenání logické 1 nebo 0 je důležité, aby Arduino zjišťovalo hodnotu napětí v době, kdy nebude docházet k překmitu. Toto nastává na rozhraní dvou bitů. Kontrola hodnoty napětí je tedy posunuta o půlku doby přechodu, tedy o 0,25 μ s. Tím je zajištěno, že Arduino nezaznamená šum vzniklý skokovou změnou referenčního signálu.

Signály do Arduina jsou posílány pomocí funkce *CallbackSerial*. Když je Arduino hotové s posunutím motorů, pošle do Matlabu jeden znak znamenající připravenost na další pokyny. V této funkci se vezmou poslední dostupné hodnoty, o kolik kroků a jakým směrem se má každý motor natočit, a pošlou se do Arduina. Tato funkce funguje i jako kontrola pro komunikaci. Bezpečnost komunikace hardwaru s počítačem je velice důležitá, jakákoli chyba v ní by znamenala, že program nebude správně fungovat. Jestliže se znak nebude shodovat s tím, který očekává Matlab, tak to bude vyhodnoceno jako chyba v komunikaci. V takovém případě se ukončí regulování polohy míčku a plošina se pomalu vrátí do své výchozí polohy. Následně je třeba program znovu inicializovat.

2.3.3 Zpracování obrazu

Poloha míčku je získávána pomocí kamery umístěné ve speciální drážce ve spodní části konstrukce. Byla vybrána kamera Symfony WC-1, která disponuje zorným úhlem až 90 °. Zároveň při jejím používání není potřeba nic instalovat, stačí pouze připojit do počítače pomocí USB kabelu. Její výhodou je i nízká cena, jak ukazuje tabulka 4 v kapitole 5. Rozlišení kamery je 1920 x 1080 pixelů, což poskytuje kvalitní obraz pro Matlab.

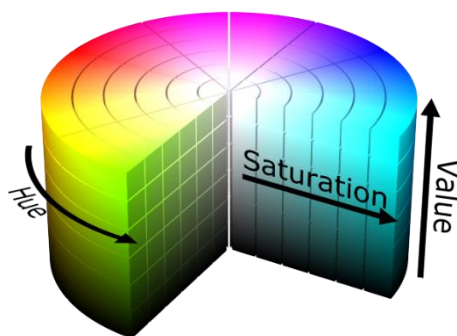
Ještě v inicializaci celého programu je možné pomocí posuvníků v grafickém prostředí upravit hodnoty pro HSV tak, aby byla kulička dobře oddělená od pozadí.

Na zpracování obrazu je možné použít několik barevných modelů. Nejznámější z nich je model RGB. Matlab umožňuje také převést snímek do odstínů šedé pomocí funkce *rgb2gray*, čímž se z něj stane černobílá verze původního obrázku. Tato funkce je součástí *Image Processing Toolbox* [16]. Z odstínů černé lze vyseparovat míček od pozadí. Pro zpracování obrazu je však použit další známý model HSV.

Na rozdíl od RGB se v tomto modelu neurčují hodnoty červené, zelené a modré, ale přímo hledaná barva. Jak ukazuje obrázek 11, celé spektrum barev je rozděleno do kružnice a pomocí parametru od 0 do 1 je možné vybrat hledaný interval. Toto je první kritérium, které se nastavuje a je označeno H (z anglického slova hue, neboli odstín).

Druhým měřítkem je saturace barvy označená S. Jak napovídá obrázek 11, tento parametr nastavuje sytost barvy. Při umístění kamery pod průhledným plexisklem se dá předpokládat, že bude potřeba odebrat světlé odstíny barev. Kamera parazitně snímá kromě plexiskla i strop místnosti, který je většinou světlý.

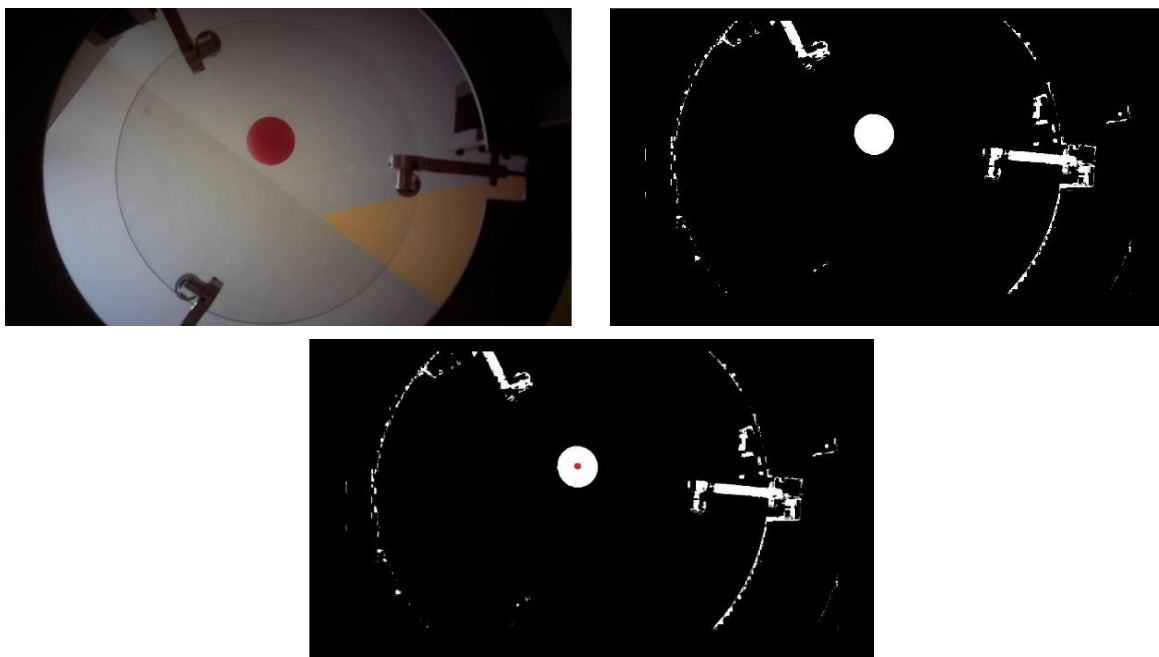
Posledním parametrem je hodnota jasu V (anglicky value). Magnetické protikusy, které spojují ramena s plexisklem, se kameře jeví jako téměř černé a mají také kruhový tvar. Je nutné nastavit hodnotu V alespoň tak, aby tyto protikusy při dalším zpracování obrazu nebyly vnímány jako spojené kruhy, ale jako nespojitě oblasti připomínající kruh.



Obrázek 11 HSV model [17]

Pro získání hodnot HSV byla použita funkce *rgb2hsv*, která je rovněž součástí *Image Processing Toolbox*. Výstupem funkce *rgb2hsv* je opět třívrstvý obraz, což je stejný formát jako známý RGB model. Dalším krokem je použití přednastavených hodnot parametrů HSV.

V Matlabu je aplikace s názvem *ColorThresholder*, umožňující přesně určit parametry HSV a z originálního snímku v levé části obrázku 12 vytvořit černobílý snímek, který je na pravé straně tohoto obrázku. Po nastavení ideálních hodnot se dá výsledek exportovat jako funkce, jejíž výstupem je již černobílý snímek vytvořený na základě nastavených parametrů daného barevného modelu. Tento snímek je zobrazený vpravo na obrázku 12. Parametry HSV je vhodné nastavovat s rezervou, neboť příliš přísná kritéria by mohla vést k nerozpoznání míčku.



Obrázek 12 Segmentace snímku

Po segmentaci míčku od pozadí je na řadě určení středu míčku. Použita byla funkce *regionprops* [16, str. 583] a její podfunkce *MinorAxisLength* a *Centroid*. Funkce *Centroid* vypočítá x-ovou a y-ovou složku středu jako průměr všech bodů v bílém obrazci. *MinorAxisLength* je použita pro kontrolu, aby Matlab pracoval pouze s obrazci, které mají velikost větší než jednotky pixelů. Na obrázku 12 dole je stejný černobílý obrázek, jako je

vpřavo, s již vyznačeným středem míčku. Střed se mnohdy nespočítá zcela dokonale, ale drobné výchyly výrazně neovlivní regulaci.

Míček může z desky spadnout nebo být odebrán uživatelem. Aby se v takovém případě deska dále nenaklápěla, byla vytvořena proměnná *not_find*, která počítá kolikrát Matlab nenašel míček na desce. Jestliže se její hodnota dostane na 20, tak se program zastaví a vrátí desku do výchozí polohy. Při každém úspěšném vyhledání míčku se tato proměnná vynuluje. Důvodem neúspěchu hledání míčku může být způsobeno také špatným nastavením hodnot HSV.

2.3.4 Naklápění roviny

Samotný náklon roviny je stěžejní částí procesu vyrovnávání míčku. V konstrukci mechanismu se mezi motory a plexisklem nachází dvě ramena, kterým je potřeba přiřadit daný úhel, aby se výsledný náklon shodoval s referenční hodnotou. Motory jsou schopny přímo ovládat pouze spodní rameno, horní se uzpůsobí mezeře mezi spodním ramenem a plexisklem.

Pro náklon desky je potřeba znát normálový vektor roviny [18], ve které má ležet kružnice reprezentující rovinu. K normálovému vektoru je nutné přiřadit další dva vektory, \vec{e} a \vec{f} , které na něj budou kolmé. Pro získání vektoru \vec{e} byla použita znalost sférických souřadnic normálového vektoru \vec{n} , která se získala ze vztahů (4).

$$r = \sqrt{x_b^2 + y_b^2 + z_b^2},$$

$$\varphi = \arctg\left(\frac{y_b}{x_b}\right),$$

$$\theta = \arccos\left(\frac{z_b}{r}\right),$$
(4)

kde souřadnice x_b, y_b, z_b jsou souřadnice bodu v kartézské soustavě a r, φ, θ jsou souřadnice bodu ve sférickém souřadném systému. Vzdálenost od počátku souřadného systému je označena r , φ značí úhel od osy x a θ úhel od svislé osy z .

Jestliže mají být vektory \vec{e}, \vec{f} kolmé na \vec{n} , pak stačí jednomu z nich přičíst k úhlu θ 90° a bude na něj kolmý. Druhý kolmý vektor se poté získá pomocí vektorového součinu. K jednotkovým vektorům \vec{e}, \vec{f} nyní stačí udělat jednotkový kruh pomocí vztahu (5).

$$k = \vec{e} \cos \gamma + \vec{f} \sin \gamma,$$
(5)

kde k je hledaná kružnice v natočeném souřadném systému a γ leží v rozmezí $\langle 0, 2\pi \rangle$. Nyní jsou známy body kružnice a stačí najít pouze ty z nich, které mají úhel φ roven úhlům umístění sférických vazeb na mechanismu (tedy $90^\circ, 210^\circ$ a 330°). Toho je docíleno opět pomocí přepočtu na sférické souřadnice a porovnání s danými hodnotami.

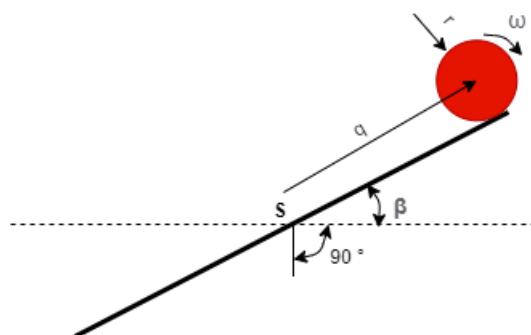
Důležitou informací je však o kolik kroků se mají posunout motory. K tomu je třeba znát jaký je úhel u spodních konců dolních ramen vůči vodorovné ose. Tento úhel vyplývá z toho, jak se ramena uspořádají v prostoru. U každého ramena je znám jeho začátek, u spodního je to hřídél motoru, u vrchního bod na plexiskle. Sestrojením dvou kružnic z těchto bodů je určen bod styku. Takové body existují vždy dva a je vybrán ten, který je blíže středu celého

mechanismu. Nyní je už možné vypočítat úhel, který svírá spodní rameno s vodorovnou rovinou a po následném přepočtu i počet kroků jednotlivých motorů.

2.3.5 Matematický model

Pro účely simulování a provádění následujících možností řízení, kterými se bude zabývat úloha pro studenty, byl vytvořen model v prostředí Simulink. Jedná se o velmi zjednodušený model simulující pouze pohyb míčku po nakloněné plošině. Pro účely této práce je takto dostatečný, ale bylo by možné do něj zahrnout i samotné krokové motory a proudy v nich. Dalšími položkami, které nejsou uvažovány jsou jakékoli odpory, ať už valivé nebo odpor vzduchu. Dalším předpokladem je trvalý kontakt míčku s plexisklem. Kontakt míčku s plexisklem by se na reálném mechanismu ztratil příliš rychlým klesáním plexiskla.

Pro odvození simulované rovnice slouží schéma na obrázku 13.



Obrázek 13 Schéma pohybu míčku

Rovnice, která se v modelu simuluje je

$$\frac{d}{dt} \frac{\partial E_k}{\partial \dot{q}} + \frac{\partial E_p}{\partial q} = 0, \quad (6)$$

kde E_k je kinetická energie míčku a E_p je potenciální energie míčku. Zobecněná souřadnice q v tomto případě značí výchylku od středu (\dot{q} pak její časovou derivaci – rychlost). Uvedená rovnice (6) vychází z metody Lagrangeových rovnic 2. druhu. Byl použit model tělesa místo hmotného bodu, což znamená, že do výpočtu kinetické energie byl zahrnut i moment setrvačnosti míčku. Kinetická energie se spočítala podle vztahu (7).

$$E_k = \frac{1}{2} m_b \dot{q}^2 + \frac{1}{2} I_b \omega^2, \quad (7)$$

kde m_b je hmotnost míčku, I_b moment setrvačnosti, který se pro kouli spočítá ze vztahu $\frac{2}{5} m_b r^2$, kde r je poloměr míčku. Posledním parametrem je ω , což je úhlová rychlost, která se počítá pomocí vztahu $\frac{\dot{q}}{r}$. Po úpravách vyjde konečný vzorec

$$E_k = \frac{7}{10} m_b \dot{q}^2. \quad (8)$$

Pro výpočet potenciální energie je použit vztah (9). Nulová hodnota potenciální energie byla zvolena jako rovnovážná poloha desky.

$$E_p = m_b g q \sin \beta, \quad (9)$$

kde g je tíhové zrychlení země. V modelu bylo zanedbáno tření mezi míčkem a deskou z toho důvodu, že i při nepatrném náklonu plexiskla došlo k rychlému pohybu míčku. Další zjednodušení rovnice vychází z toho, že úhel β je malý čili $\sin \beta \approx \beta$. Výsledná rovnice, která byla naprogramována do Simulinku má tvar

$$\frac{7}{5} \ddot{q} + g \beta = 0, \quad (10)$$

kde \ddot{q} charakterizuje zrychlení míčku. Jak je patrné z rovnice (10), tak nezáleží na hmotnosti míčku, ani na jeho poloměru. Maticový zápis této rovnice je ve tvaru

$$\begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{5}{7} g \sin \beta \end{bmatrix}. \quad (11)$$

2.4 Grafické prostředí

K celému mechanismu bylo vytvořeno i grafické prostředí (General User Interface = GUI), které usnadní ovládání a nastavování ostatním uživatelům. Toto prostředí je naprogramováno pomocí objektově orientovaného programování. Na obrázku 14 je zobrazen vzhled tohoto prostředí se všemi náležitostmi popsány v následujících odstavcích.

V levé horní části jsou vypsány kamery, které jsou připojeny k počítači. Jedna z nich je i ta, která má být použita při řízení. V blízkosti se nachází i tlačítko „refresh cam“ pro případ, že kamera, která má být použita, ještě nebyla připojena k počítači. Toto tlačítko tedy znovu najde veškeré kamery připojené k zařízení. Po vybrání konkrétní kamery se ve spodní části objeví dva obrázky toho, co kamera vidí ve skutečnosti a výstup po zpracování obrazu ve formě binárního snímku.

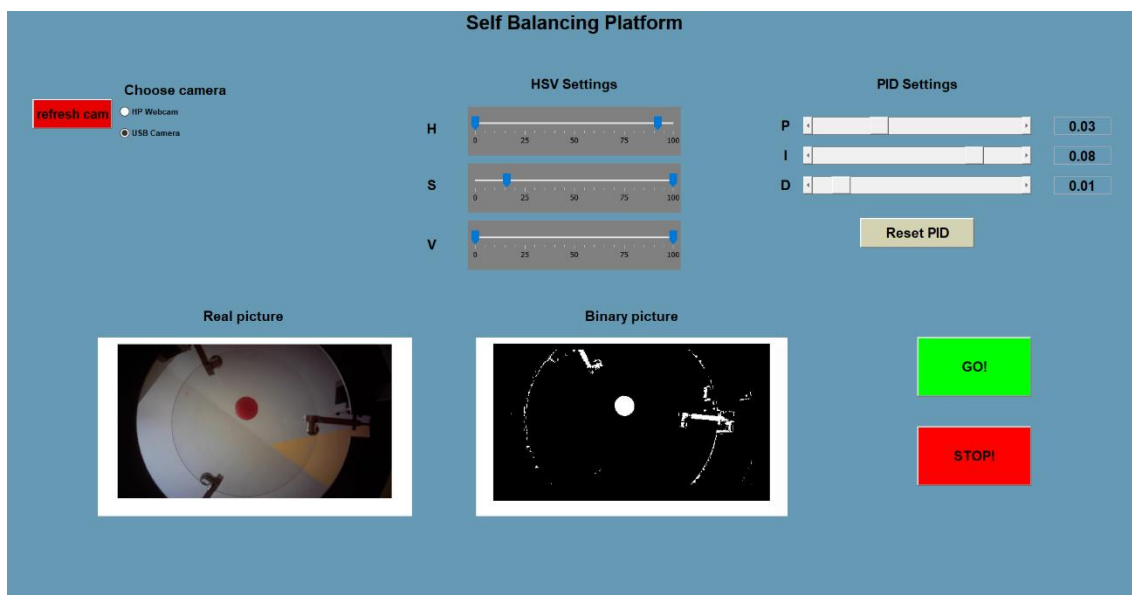
V další části jsou nastavovány hodnoty HSV pro zpracování obrazu. Tyto slidery nenabízí přímo Matlab, ale jelikož se do GUI mohou vkládat komponenty z jazyku Java, tak bylo umožněno utvořit slider se dvěma posuvníky. Ve spodní části se po každém posunutí těmito posuvníky aktualizuje skutečný obraz kamery se svým novým výstupem po zpracování obrazu.

Vedle sliderů pro HSV se nachází další slidery sloužící pro nastavení hodnot PID regulátoru. Tyto hodnoty se mohou zadat buď posouváním na slideru, nebo přímo napsáním hodnoty do pole po jejich pravé straně. Tyto komponenty tvoří provázané dvojice, takže změna na jednom z nich způsobí změnu i na tom druhém.

V pravém dolním rohu se nachází dvě tlačítka, zelené „GO!“, které spustí vyvažování míčku s parametry, které byly zadány pomocí sliderů. Červeným tlačítkem „STOP!“ se program zastaví. Matlab dá zvukové znamení, že toto tlačítko bylo zmáčknuté, komunikace s Arduinem se přeruší a uživatelské prostředí se zavře.

Jestliže dojde v programu k nějaké chybě, tak se pod tlačítkem „STOP!“ objeví okno, které vypíše chybovou hlášku.

Celé toto grafické prostředí bylo tvořeno s důrazem na jednoduchost ovládání a srozumitelnost. Jeho rozšíření by mohlo zahrnovat i nastavení samotné kamery a další pokročilejší parametry.



Obrázek 14 Grafické prostředí

3 Úloha pro studenty

Tato úloha byla vytvořena pro účely rozšíření znalostí studentů nižších ročníků. Důvod, proč řešit tuto úlohu, je zdokonalit stávající program. Budou změněny celkem 3 části, z nichž jedna se odehrává v Simulinku. Pro řešení této úlohy je dostačující mít základní povědomí o programování v Matlabu a maticových operacích.

3.1 Zadání

- 1) Změnit způsob zpracování obrazu - místo funkce *regionprops* použít Houghovu kruhovou transformaci [19].
- 2) Použít transformační matice pro výpočet bodů A2, B2 a C2 z obrázku 5.
- 3) Implementovat jiné řízení než pomocí PID regulátoru v matematickém modelu v Simulinku.

3.2 Zpracování obrazu

Pro tuto část úlohy je potřeba pořídit snímek, na kterém se bude úloha provádět. Po připojení kamery k počítači se použije příkaz *snapshot*, který vytvoří snímek. Získaný obrázek se poté zobrazí pomocí příkazu *imshow*.

V dosavadním řešení programu se střed kuličky určuje pomocí metody *centroids*. Princip této funkce byl vysvětlen v kapitole 2.3.3. Existuje několik různých metod na určování středu kruhových obrazců a v této úloze bude demonstrována Houghova kruhová transformace.

Prvním krokem bude nalezení míčku na snímku. K tomu je možné použít dosavadní postup nebo přímo funkci *edge*, která je v Matlabu vestavěná. Při použití této funkce je na výběr několik metod, kterými se má obraz zpracovat. Jmenovitě například *Canny*, *Sobel* a *Log* [20]. Nejvhodnější pro tuto úlohu bude metoda *Canny*. *Canny* je málo náchylná na šum a dá dobrý základ pro použití Houghovy transformace.

Pro řešení této části je podstatné uvědomit si, že se na snímku hledá kruhový objekt. Každý takový objekt se dá popsat rovnicí

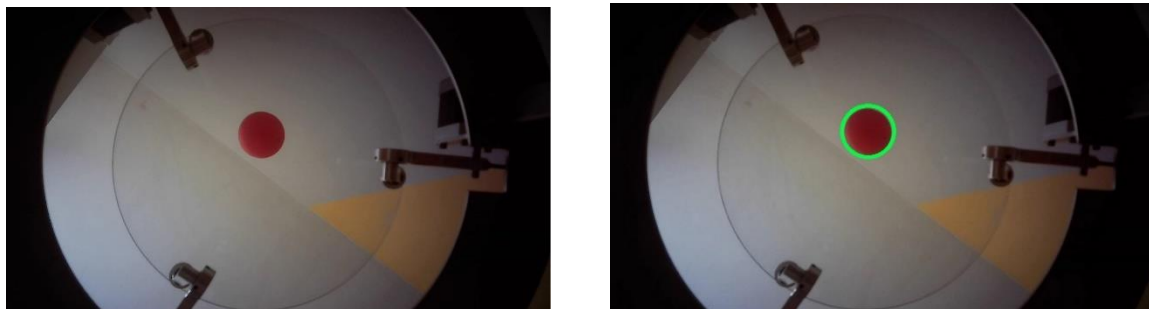
$$(x - x_b)^2 + (y - y_b)^2 = r^2, \quad (12)$$

kde x_b , y_b tvoří x-ovou a y-ovou souřadnici středu, r je poloměr kružnice a x , y jsou souřadnice bodů vzniklé kružnice.

Problém nastává v určení poloměru kružnice. Lze konstatovat, že míček bude mít na každém snímku rozdílný poloměr. To je způsobeno odrazem světla od povrchu míčku, což způsobuje šum. Je tedy nutné vybrat interval poloměrů, kterých může míček nabývat. Je žádoucí, aby tento interval byl co nejmenší.

Principem Houghovy kruhové transformace je to, že z každého bodu, který je označen jako okraj, je vytvořena kružnice podle rovnice (12). Vznikne množina bodů, z nichž každý může být středem hledané kružnice. Poté, co se tento postup zopakuje pro každý bod, vznikne proměnná velkých rozměrů s možnými středy kružnice. Jeden bod bude společný všem těmto kružnicím, a ten bude prohlášen za střed vyšetřované kružnice.

Houghova kruhová transformace se v Matlabu provádí velice snadno, jedná se opět o vestavěnou funkci *imfindcircles* [16]. Jejím vstupem jsou obraz, ve kterém se mají hledat kruhy a rozpětí poloměrů. Dalším vstupem může být i citlivost, ale není nutné tento parametr definovat. Výstupem funkce jsou středy a poloměry všech kružnic, které funkce našla. Na obrázku 15 je vidět porovnání originálního snímku a výstupu funkce *imfindcircles* s vyznačeným okrajem míčku.



Obrázek 15 Určení středu míčku Houghovou transformací

3.3 Transformační matice

Následující část je věnovaná způsobu výpočtu souřadnic bodů A2, B2 a C2 z obrázku 5. V současné verzi se využívá normálového vektoru, ze kterého se získávají souřadnice těchto bodů. Pomocí znalostí z kinematiky je zřejmé, že k určení stačí použít transformační matice. Implementováním těchto matic dojde ke zrychlení programu. Transformační matice v prostoru jsou 3 [21].

$$\begin{aligned} R_{\psi} &= \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\ R_v &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos v & -\sin v \\ 0 & \sin v & \cos v \end{bmatrix}, \\ R_{\phi} &= \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (13)$$

Úhly v těchto transformačních maticích se nazývají Eulerovy úhly. Pro pochopení principu budou uvažovány dva totožné souřadné systémy označené c , d . Jestliže se souřadný systém d otočí kolem své osy z_d , tak úhel, který vznikne mezi souřadnými systémy, se nazývá precese a je označen písmenem ψ . Následně, pokud se souřadný systém d natočí kolem své současné osy x_d , tak se vzniklý úhel nazývá nutace a je označen písmenem v . Natočením kolem současné osy z_d souřadného systému d vznikne úhel označený písmenem ϕ , který se nazývá rotace. Pomocí zmíněných Eulerových úhlů je možné těleso jakkoli orientovat v prostoru. Výsledná matice R má následně tvar

$$R = R_{\psi} R_v R_{\phi}. \quad (14)$$

Matlab umožňuje usnadnění tohoto zápisu, aby se nemusely vypisovat matice ze vztahu (13) ručně. Stačí zadat například *rotx(10)*, což značí rotaci kolem osy x o 10° . Obdobně je možné postupovat i při rotaci kolem osy z .

Jelikož matice \mathbf{R} vznikla maticovým součinem, který není komutativní, je nutné dodržet pořadí matic při násobení, jak je ukázáno ve vztahu (14). Pro výpočet souřadnic bodů A2, B2 a C2 nyní stačí dosadit do rovnice $A'_2 = \mathbf{R}A_2$, kde A'_2 jsou nové souřadnice bodu a A_2 jsou souřadnice ve vodorovném stavu.

3.4 Řízení

Další částí této úlohy bude upravit řízení celého mechanismu. V aktuální verzi je řízení zajištěno pomocí PID regulátoru s vhodně vybranými koeficienty. V této pasáži bude demonstrován pokročilejší způsob řešení řízení. Využita bude maticová rovnice (11), která je v Simulinku již připravená.

Ke stávající rovnici je třeba doplnit ještě jednu rovnici, výstupovou. (Ta, která je již nasimulovaná, se nazývá stavová.) Stavovou rovnici je tedy možné napsat v maticové podobě jako

$$\dot{x} = \mathbf{A}x + \mathbf{B}u, \quad (15)$$

kde x je vektor stavů a u je vektor vstupů. \mathbf{A} a \mathbf{B} jsou vyznačeny tučně, protože znázorňují matice. Výstupová rovnice má tvar

$$y = \mathbf{C}x + \mathbf{D}u, \quad (16)$$

kde y jsou výstupy z rovnice (výchylka a rychlost). Matice \mathbf{D} bude rovna 0 a matice \mathbf{C} je pro určení stavu, který má být řízen. Jelikož má být regulována poloha, tak bude mít matice \mathbf{C} tvar

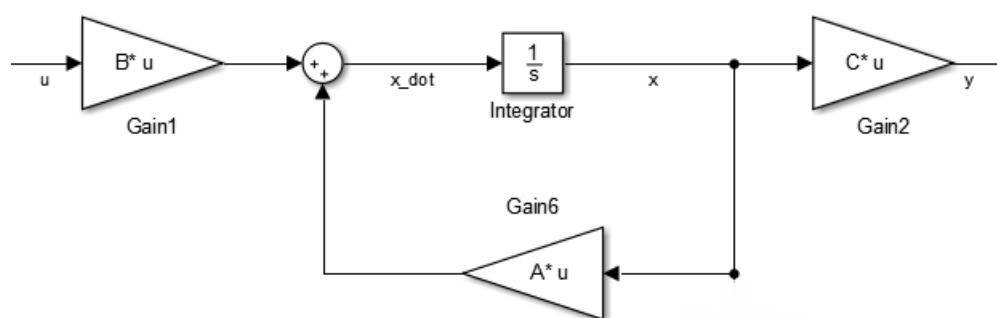
$$\mathbf{C} = [1 \ 0].$$

Ostatní matice mají tvar

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},$$

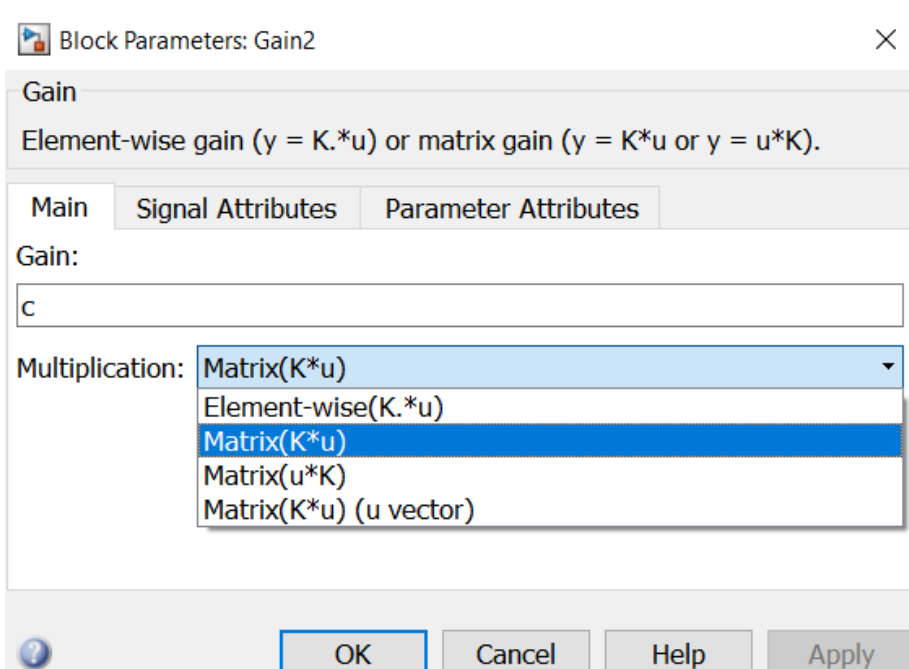
$$\mathbf{B} = \begin{bmatrix} 0 \\ -\frac{5}{7}g \sin \beta \end{bmatrix}.$$

V Simulinku bude následně vytvořen stavový model, který bude odpovídat obrázku 16.



Obrázek 16 Schéma v Simulinku

Všechny bloky Gain je potřeba nastavit na maticové násobení. Po jejich rozkliknutí se objeví dialogové okno, kde musí být zvolena možnost Matrix ($\mathbf{K} \cdot u$), jak ukazuje obrázek 17.



Obrázek 17 Nastavení bloku Gain

Na obrázku 18 je zobrazeno schéma se zpětnovazebným řízením realizovaným maticí \mathbf{K} . Tato matice, přesněji řádkový vektor, je řídicí člen s koeficienty určenými podle metody Pole Placement. V této úloze budou koeficienty vektoru \mathbf{K} určeny pomocí *Ackermann's Formula* [22]. Výstupová rovnice při použití metody Pole Placement nezmění svůj tvar. Nová podoba stavové rovnice, která přímo vyplývá ze obrázku 18, je

$$\dot{x} = (\mathbf{A} - \mathbf{BK})x. \quad (17)$$

Použití této metody má za podmínku říditelnost systému (controlability). Matice říditelnosti \mathbf{M} se v Matlabu určí snadno pomocí funkce *ctrb*, jejímiž vstupy jsou matice \mathbf{A} a \mathbf{B} . Tato matice se určí podle vztahu (18).

$$\mathbf{M} = [\mathbf{B} \quad \mathbf{AB}]. \quad (18)$$

Hodnost této matice (příkaz *rank*) musí být stejná, jako řád systému, který je v tomto případě roven 2. Řád systému je dán maximálním řádem derivace ve stavové rovnici povýšen o 1.

Dalším krokem je určení požadovaných pólů systému λ_1 a λ_2 . Všechny póly musí být záporné a jejich velikost může být variabilní. Po dosazení do rovnice (19) vznikne polynom druhého řádu s koeficienty α_1 a α_2 .

$$|s\mathbf{I} - \mathbf{A} + \mathbf{BK}| = (s - \lambda_1)(s - \lambda_2) = s^2 + \alpha_1 s + \alpha_2, \quad (19)$$

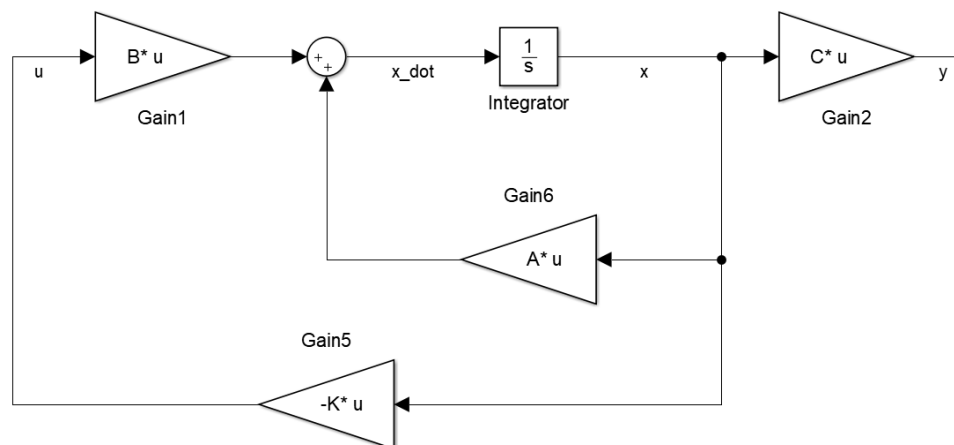
kde \mathbf{I} je jednotková matice a s je komplexní proměnná. Pro určení vektoru \mathbf{K} je ještě nutné získat koeficienty α_1 a α_2 , které se vypočítají ze vztahu (20).

$$|s\mathbf{I} - \mathbf{A}| = s^2 + a_1 s + a_2. \quad (20)$$

Ze znalosti těchto koeficientů je už možné určit vektor \mathbf{K} ze vztahu (21).

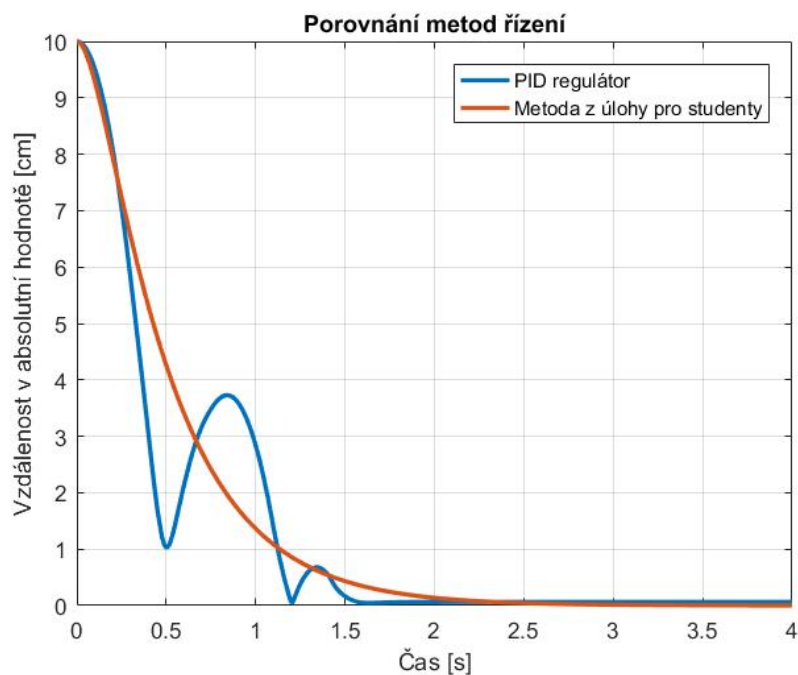
$$\mathbf{K} = [\alpha_2 - a_2; \alpha_1 - a_1]. \quad (21)$$

Je zřejmé, že se jedná o dvousložkový vektor, jehož složky se vložily do příslušného bloku Gain v Simulinku na obrázku 18.



Obrázek 18 Zpětnovazební řízení modelu

Obrázek 19 porovnává použití PID regulátoru a metody, která byla popsána v předcházejících odstavcích. Oba modely jsou značně zjednodušené, jejich účelem je pouze toto srovnání. Do PID regulátoru byly dosazeny hodnoty odpovídající skutečnému mechanismu. Na obrázku 19 je vidět, že metoda stavového řízení dostane míček do nulové vzdálenosti od požadovaného bodu přesně bez přestřelení. U PID regulátoru dojde i k drobnému přestřelení, proto se křivka na chvíli vzdaluje od požadované hodnoty.



Obrázek 19 Srovnání obou metod

4 Výsledky

V této kapitole budou rozebrány výsledky vývoje mechanismu. Použití krokových motorů bylo zvoleno za účelem přesného náklonu plexiskla. Motory ale neovlivňují náklon přímo, proto byl proveden experiment, který zkoumá přesnost náklonu. Další část se zabývá samotným regulováním polohy míčku jak v matematickém modelu, tak i na reálném mechanismu. V poslední části této kapitoly je zobrazen výsledný vzhled mechanismu.

4.1 Přesnost náklonu

Krokové motory patří mezi pohony, které nepotřebují mít regulátory polohy. Výsledný náklon plexiskla je však ovlivněn vřely ve vazbách a provedenými zjednodušeními ve výpočtu. Motory se pohybují po $0,9^\circ$, což může také vést k nepřesnostem. Cílem experimentu je zjištění přesnosti náklonu. Měření bylo provedeno digitálním posuvným měřidlem s rozlišitelností 0,01 mm podle schématu na obrázku 20.



Obrázek 20 Schéma experimentu

Byla měřena vzdálenost y a z ní dopočítáván úhel Ω pomocí vztahu (22)

$$\Omega = \sin^{-1} \frac{l}{r} = \sin^{-1} \frac{h - y}{r}, \quad (22)$$

kde h je výška rovnovážné polohy od roviny s motory (109,22 mm), r je poloměr plexiskla (12 cm) a l je rozdíl $h - y$. Měření bylo provedeno v jedné ose po kroku 2° a každá hodnota byla měřena desetkrát.

Pro výpočet standardní nejistoty typu A byl použit vzorec (23)

$$U_A = \frac{s_l}{\sqrt{n}} \quad (23)$$

kde U_A je hledaná nejistota, n je počet měření a s_l je výběrová směrodatná odchylka daná vztahem (24).

$$s_l = \sqrt{\frac{\sum (l_i - \bar{l})^2}{n - 1}}, \quad (24)$$

kde l_i je hodnota i -tého měření a \bar{l} je aritmetický průměr naměřených délek.

Jelikož cílem bylo určit úhel náklonu místo naměřené vzdálenosti, je nutné výsledky zpracovat jako nepřímou měřenou veličinu. Střední hodnotou tohoto úhlu bude opět hodnota vypočtená podle vztahu (22). Nejistota tohoto úhlu se poté vypočítá ze vztahu

$$U_\Omega = \frac{d}{dl} \left(\sin^{-1} \frac{l}{r} \right) \cdot U_A = \frac{1}{\sqrt{r^2 - l^2}} U_A. \quad (25)$$

Celkové výsledky tohoto experimentu jsou uvedeny v tabulce 3.

Tabulka 3 Výsledky experimentu

	Požadovaný úhel náklonu [°]									
	-10	-8	-6	-4	-2	2	4	6	8	10
l_t [mm]	130,07	125,93	121,77	117,60	113,41	105,04	100,86	96,68	92,53	88,39
\bar{l} [mm]	129,81	125,63	121,62	117,64	113,37	105,06	100,84	96,72	92,69	88,54
s [mm]	0,20	0,24	0,28	0,25	0,30	0,27	0,30	0,27	0,27	0,31
U_A [mm]	0,06	0,07	0,09	0,08	0,09	0,08	0,09	0,09	0,09	0,10
Ω [°]	-9,8794	-7,8579	-5,9331	-4,0226	-1,9795	1,9890	4,0063	5,9787	7,9167	9,9235
U_Ω [°]	0,0010	0,0011	0,0011	0,0010	0,0011	0,0009	0,0009	0,0008	0,0008	0,0009

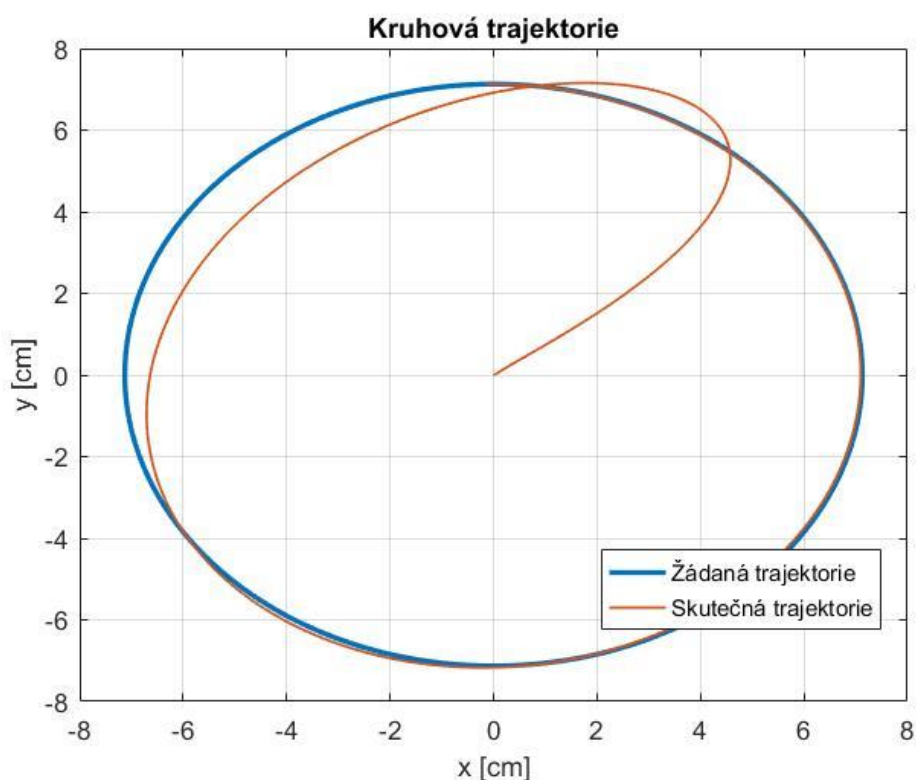
V prvním řádku tabulky jsou požadované hodnoty úhlů. Druhý řádek vyjadřuje vzdálenost l_t , což je teoretická hodnota pro daný úhel podle vztahu (22). Na dalším řádku jsou průměrné hodnoty délek \bar{l} získané experimentem. Na posledních dvou řádcích jsou statisticky zpracované hodnoty úhlu Ω i se svojí nejistotou měření U_Ω . Výsledky po rozšíření o své nejistoty nezahrnují požadovaný úhel, ale ve velké míře se mu blíží. Vzniklé rozdíly jsou zanedbatelné.

Zároveň stojí za povšimnutí fakt, že aritmetický průměr naměřených délek \bar{l} na třetím řádku tabulky 3 se téměř shoduje s teoretickými hodnotami. Čím menší v absolutní hodnotě byl požadovaný úhel, tím více se výsledky shodují. Tímto experimentem bylo dokázáno, že plošina se díky přesným krokovým motorům dokáže natočit téměř přesně pod požadovaným úhlem.

4.2 Ukázky regulace

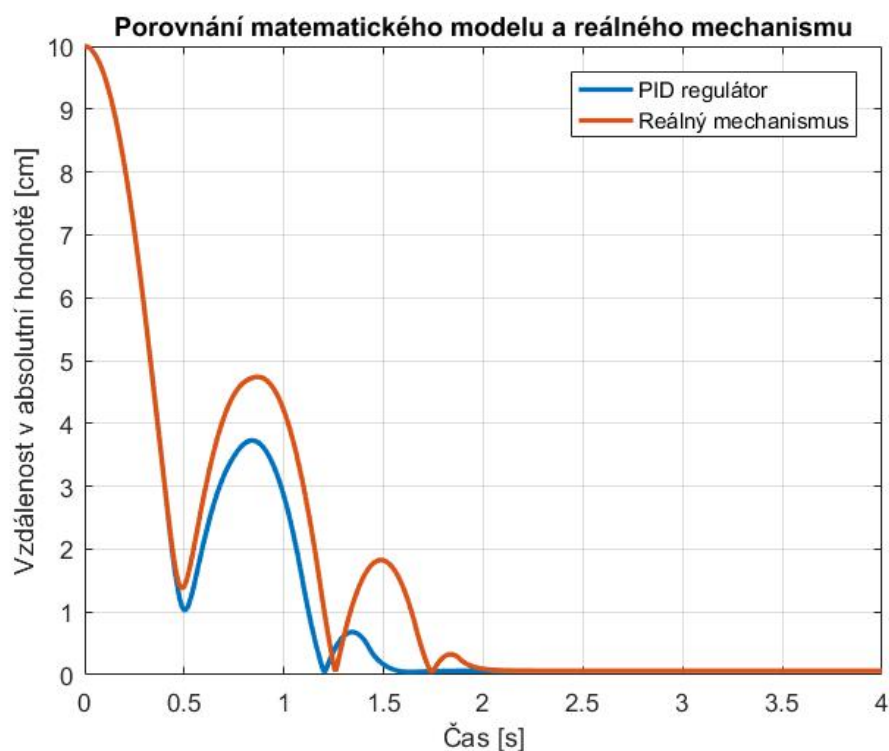
V této části budou demonstrovány výsledky regulování jak matematického modelu, tak reálného zařízení.

Na obrázku 21 je zobrazena regulace matematického modelu po žádané kruhové trajektorii. Míček měl na počátku regulace souřadnice [0,0] a nulovou počáteční rychlost. Simulace trvala pouze po dobu, kdy by se vykonal pouze jeden a čtvrt oběhu kružnice, neboť by následně obě křivky splynuly do sebe. Po uplynutí jedné otáčky už poloha míčku přesně korespondovala se žádanou polohou.



Obrázek 21 Kruhová trajektorie v modelu

Na obrázku 22 je poté zobrazena regulace na skutečném zařízení s požadavkem nulové odchylky od středu plexiskla.



Obrázek 22 Regulace polohy míčku

Z obrázku 22 jsou patrné nedokonalosti způsobené zjednodušením matematického modelu. Všechna provedená zjednodušení vedla k většímu přestřelení požadované hodnoty (vyšší odchylka při vzdalování) a k jejich většímu počtu. Poloha míčku se však ustálila na požadované hodnotě v relativně krátkém čase.

4.3 Výsledný vzhled

Výsledný vzhled zařízení je zobrazen na obrázku 23. Kabeláž motorů je vedena svislými stojnami do spodní části konstrukce, kde je připojena k Arduino. Z dolní části konstrukce jsou vyvedeny 3 kabely. Modrý stíněný je připojen k Arduino. Jeho druhý konec musí být připojen do předem definovaného portu v počítači. Černý kabel je od kamery, u něj nezáleží, do kterého portu je připojen. Matlab by měl připojenou kameru najít v jakémkoli portu.

Na odvrácené straně je vyveden síťový kabel, který je připojen na externí zdroj napájející CNC Shield. Tento kabel je cíleně co nejdál od kabelu připojenému k Arduino. Důvodem je možné rušení komunikace mezi Arduinem a Matlabem.

Na spodní části konstrukce se nacházejí dva vypínače na odpojení části mechanismu od sítě. První odpojí externí zdroj od sítě, druhý odpojí výstup zdroje od krokových motorů. Součástí mechanismu je i jistič, který ochrání uživatele před případným nepředvídatelným stavem, který by ho mohl ohrozit.



Obrázek 23 Výsledný vzhled

5 Přehled nákladů

Při vymýšlení tohoto mechanismu byl kladen důraz na ekonomickou stránku. Jsou použity pouze 3 motory, které zajišťují dostatečnou stabilitu a tuhost. V tabulce 4 jsou uvedeny náklady, které byly vynaloženy za jednotlivé komponenty. Motory patří k jednomu z nejnákladnějších položek, proto byla snaha o minimalizaci počtu. Navíc každý motor potřebuje i svůj driver.

Tabulka 4 Shrnutí částek

Název	Množství	Cena / 1 ks	Cena za položku
Motor Nema 17HS4401	3	318 Kč	954 Kč
Driver DRV8825	3	68 Kč	204 Kč
CNC Shield	1	78 Kč	78 Kč
Magnetické úchyty	3	28 Kč	84 Kč
Plexisklo	1	85 Kč	85 Kč
Arduino Mega 2560	1	428 Kč	428 Kč
USB - B kabel	1	28 Kč	28 Kč
Kamera	1	899 Kč	899 Kč
Spínač 250V/16A	1	56 Kč	56 Kč
Spínač 250V/6A	1	16 Kč	16 Kč
Síťový kabel	1	76 Kč	76 Kč
Jistič	1	111 Kč	111 Kč
Suma	-	-	3 019 Kč

V tabulce 4 nejsou zahrnuty náklady na výrobu celé konstrukce ukrývající elektroniku s Arduinem a podporující motory. Důvod je ten, že tato konstrukce byla vyrobena zdarma. Jsou zde tedy zobrazeny pouze skutečné náklady, při výrobě většího množství kusů by konstrukce už zdarma nebyla a musela by se zahrnout do ekonomické stránky vývoje.

Závěr

V rešeršní části práce jsou popsány možné konstrukce a provedení popisovaného mechanismu. Byly zmíněny pouze ty varianty, které jsou relevantní vzhledem k této práci.

Hlavním cílem této práce bylo vymyslet konstrukci a program, což bylo splněno. Mechanismus pracuje správně a je schopen umístit míček na požadované místo. K mechanismu byla vytvořena i úloha pro studenty nižších ročníků, pro procvičení metod řízení a zpracovávání obrazu z kamery. Posledním úkolem bylo vytvoření matematického modelu v nástroji Simulink. Jedná se o velmi zjednodušený model, avšak pro účely této práce je zcela dostačující. Slouží pouze pro porovnání užití PID regulátoru a metody popisované v úloze pro studenty.

Bylo vytvořeno také grafické uživatelské prostředí, které umožňuje nastavovat parametry použité v programu. Toto prostředí slouží uživatelům, kteří by nerozuměli samotnému programu. Pro schopnost jeho ovládání stačí základní povědomí o PID regulátoru a znalost barevného HSV modelu.

Vlastní mechanismus je vybaven řadou ochran, které zajišťují jeho správné fungování. Kdykoli je jedna z těchto ochran splněna, vrátí motory plexisklo zpět do výchozí pozice a program musí být znovu inicializován. Jedná se o ochrany správné komunikace hardwaru s počítačem a kontrola hledání míčku.

Další možnosti vývoje mohou být následující:

- Vlastní plošný spoj: V této práci bylo použito Arduino Mega 2560 z důvodu úspory času. Dalším důvodem jeho použití byl fakt, že hlavním cílem práce nebylo vymyslet vlastní tištěný spoj.
- Úprava trajektorie míčku: Program je zatím vytvořen pro regulaci polohy míčku na střed plexiskla. Jelikož mechanismus dokáže dobře regulovat polohu míčku, tak je možné přejít k pokročilejším možnostem, jako je například trajektorie kružnice, čtverce a jiných geometrických tvarů. Zvedáním plexiskla ve svislém směru je možné docílit i kontrolovaného skákání míčku.

Seznam použité literatury

- [1] Drony v armádě. In: DRONPRO [online]. [cit. 2021-03-29]. Dostupné z: <https://dronpro.cz/drony-v-armade-a-u-policie>
- [2] ALSAMARJI, Jihad. Design and Control of a Ball and Plate Didactic Device. Brusel, 2019. Diplomová práce. Université Libre de Bruxelles. Vedoucí práce Prof. Dr. Ir. Michael Kinnaert.
- [3] Motion Systems [online]. Varšava, 2009 [cit. 2021-03-04]. Dostupné z: <https://motionsystems.eu>
- [4] Lineární motor. In: LINEÁRNÍ MOTORY [online]. [cit. 2021-03-22]. Dostupné z: <http://www.linearni-motor.cz>
- [5] Servomotory. In: SEW EURODRIVE [online]. [cit. 2021-03-07]. Dostupné z: https://www.seweurodrive.cz/vyrobky/motory/servomotory/servomotoren_3.html
- [6] SERVOMOTORY. In: RAVEO [online]. [cit. 2021-03-07]. Dostupné z: https://www.raveo.cz/servo_motor?fbclid=IwAR2Y2saQY2x9rgSLLS7VCwM0oExsye2QevPjpqEWcM5zFKTXzvvt0XKrbG0
- [7] Arduino Mega 2560 Datasheet. In: RobotShop: RobotShopPutting robotics at your service [online]. [cit. 2021-04-05]. Dostupné z: <https://www.robotshop.com/media/files/pdf/arduinomega2560datasheet.pdf>
- [8] 3-Axis CNC/Stepper Motor Shield for Arduino. In: HandsOn Tech: Open Source Electronics Platform [online]. [cit. 2021-04-05]. Dostupné z: <https://www.handsontec.com/dataspecs/cnc-3axis-shield.pdf>
- [9] DRV8825 Stepper Motor Controller IC. In: Texas Instruments [online]. 2014 [cit. 2021-04-05]. Dostupné z: https://www.ti.com/lit/ds/symlink/drv8825.pdf?ts=1617587131436&ref_url=https%253A%252F%252Fwww.ti.com%252F
- [10] DRV8825 Stepper Motor Driver Carrier. In: Pololu [online]. [cit. 2021-03-14]. Dostupné z: <https://www.pololu.com/product/2987>
- [11] Odis Harkins. Re: How do I split an incoming string [příspěvek v diskuzním fóru]. In: stackExchange [online]. 19.4.2014, 19:08 Dostupné z: <https://arduino.stackexchange.com/questions/1013/how-do-i-split-an-incoming-string>
- [12] 17HS4401 Datasheet – 40mm, Stepper Motor. In: DatasheetCafe [online]. [cit. 2021-02-17]. Dostupné z: <http://www.datasheetcafe.com/17hs4401-datasheet-stepper-motor/>
- [13] KROKOVÉ MOTORY. In: PROFIROBOT [online]. [cit. 2021-02-17]. Dostupné z: <http://profirobot.cz/wp-content/uploads/2018/03/Krokové-motory.pdf>
- [14] Microstepping myths. In: MachineDesign [online]. [cit. 2021-04-05]. Dostupné z: <https://www.machinedesign.com/archive/article/21812154/microstepping-myths>
- [15] Read from and write to a serial port. Web.dev [online]. 12.8.2020 [cit. 2021-02-17]. Dostupné z: <https://web.dev/serial/>.

- [16] Image Processing Toolbox User's Guide [online]. 11.2. 2020 [cit. 2021-02-17]. Dostupné z: https://www.mathworks.com/help/pdf_doc/images/images Ug.pdf
- [17] Grafické zobrazení HSV. In: Wikipedie [online]. [cit. 2021-03-22]. Dostupné z: <https://cs.wikipedia.org/wiki/HSV>
- [18] Rumburak. Re: kružnice v prostoru [příspěvek v diskuzním fóru]. In: *Matematické forum* [online]. 24.06.2011, 15:33. Dostupné z: <https://forum.matematika.cz/viewtopic.php?pid=211567>
- [19] JAGADALE, Amol B a D. V JADHAV. Early detection and categorization of cataract using slit-lamp images by hough circular transform. In: 2016 International Conference on Communication and Signal Processing (ICCSP) [online]. IEEE, 2016, s. 0232-0235 [cit. 2021-03-22]. Dostupné z: doi:10.1109/ICCSP.2016.7754129
- [20] Edge detection and mathematic fitting for corneal surface with Matlab software. International Journal of Ophthalmology [online]. 2017 [cit. 2021-02-16]. ISSN 22223959. Dostupné z: doi:10.18240/ijo.2017.03.02
- [21] GREPL, Robert. *Kinematika a dynamika mechatronických systémů*. Brno: Akademické nakladatelství CERM, 2007. ISBN 978-80-214-3530-8.
- [22] HSU, Lee-chuang a Fan-ren CHANG. The generalized Ackermann's formula for singular systems. *Systems & control letters* [online]. Elsevier B.V, 1996, 27(2), 117-123 [cit. 2021-5-19]. ISSN 0167-6911. Dostupné z: doi:10.1016/0167-6911(95)00065-8

Seznam použitých symbolů

Symbol	Název	Jednotka
a, α	koeficienty metody řízení	-
A	ampér	-
A, B, C, D	stavové matice	-
$A1, B1, C1$	hřídele motorů	-
$A2, B2, C2$	sférické vazby u plexiskla	-
$A3, B3, C3$	klouby mezi rameny	-
c, d	označení souřadných systémů	-
$\vec{e}, \vec{f}, \vec{n}$	vektory	-
E_k	kinetická energie	J
E_p	potenciální energie	J
f_i	počet stupňů volnosti vazeb	-
g	gravitační zrychlení	m/s ²
GUI	General User Interface	-
H	Hue = odstín	-
h, l, y	vzdálenost	cm
HSV, RGB	barevné modely	-
i	počet stupňů volnosti	-
I	jednotková matice	-
I_b	moment setrvačnosti	kgm ²
j	počet kinematických dvojic	-
K	řídící člen	-
\bar{l}	aritmetický průměr	mm
l_i	i-té měření	mm
M	matice říditelnosti	-
m_b	Hmotnost míčku	kg
n	počet měření	-
n_t	počet těles	-
Ncm	moment	-
P	střed plošiny (požadovaný bod)	pixel
q	zobecněná souřadnice	cm
r	poloměr	cm
R_ψ, R_v, R_ϕ	matice rotace	-
s	komplexní proměnná	-
s_l	výběrová směrodatná odchylka	mm
S	Saturation = saturace	-
S_b	střed míčku v pixelech	pixel
t	čas	s
u	vstupní vektor	-
U_A	nejistota typu A	mm
U_Ω	nejistota měření úhlu	°
V	volt	-

V	Value = jas	-
x	stavový vektor	-
X, Y, Z	sloty pro drivery	-
x_b, y_b, z_b	souřadnice bodu	cm
x_d, z_d	osy x a z souřadného systému d	-
x_p	vzdálenost v pixelech	pixel
$\beta, \gamma, \theta, \varphi, \phi, \psi, \Omega$	úhel	°
λ_1, λ_2	požadované póly systému	-
μs	mikrosekunda	-
ω	úhlová rychlost	rad/s

Seznam obrázků

Obrázek 1 Více variant mechanismu	12
Obrázek 2 Výrobek PS-3TM3-200 [3]	13
Obrázek 3 Řez lineárním motorem [4]	14
Obrázek 4 Nákres konstrukce	16
Obrázek 5 Schéma mechanismu	16
Obrázek 6 Soustava ramen	18
Obrázek 7 Arduino Mega 2560 [7]	18
Obrázek 8 CNC Shield [8]	19
Obrázek 9 Driver DRV8825 [10]	20
Obrázek 10 Schéma programu	22
Obrázek 11 HSV model [17]	25
Obrázek 12 Segmentace snímku	25
Obrázek 13 Schéma pohybu míčku	27
Obrázek 14 Grafické prostředí	29
Obrázek 15 Určení středu míčku Houghovou transformací	31
Obrázek 16 Schéma v Simulinku	32
Obrázek 17 Nastavení bloku Gain	33
Obrázek 18 Zpětnovazební řízení modelu	34
Obrázek 19 Srovnání obou metod	34
Obrázek 20 Schéma experimentu	35
Obrázek 21 Kruhová trajektorie v modelu	37
Obrázek 22 Regulace polohy míčku	37
Obrázek 23 Výsledný vzhled	39

Seznam tabulek

Tabulka 1 Parametry motoru	21
Tabulka 2 Ztráty výkonu mikrokrokováním	22
Tabulka 3 Výsledky experimentu	36
Tabulka 4 Shrnutí částek	40

Seznam příloh

- 1) Výkresová dokumentace
- 2) General User Interface
- 3) Matematický model
- 4) Finální program regulace